

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

REALD SPARK, LLC,
Patent Owner.

Case No. IPR2023-01046

Patent No. 10,740,985

DECLARATION OF LINA KARAM, PH.D.

U.S. PATENT NO. 10,740,985 (CLAIMS 1-14 AND 16-19)

Petitioner Ex. 1003

Contents

I.	ENGAGEMENT	1
II.	PROFESSIONAL BACKGROUND	1
III.	MATERIALS REVIEWED	10
IV.	DESCRIPTION OF THE RELEVANT TIMEFRAME, THE RELEVANT FIELD, AND A PERSON OF ORDINARY SKILL IN THE ART	10
V.	TECHNOLOGY BACKGROUND	11
A.	Machine Learning: <i>General</i>	11
B.	Supervised Learning vs Unsupervised Learning.....	13
C.	Objective Function, Training, Testing, and Deployment.....	14
D.	Common Machine Learning Algorithms and Architectures	16
1.	<i>Neural Networks</i>	16
2.	<i>Random Forests</i>	24
3.	<i>Teacher-Student Architecture</i>	29
E.	Gaze Correction	31
F.	Optical Flow and Warping	34
VI.	OVERVIEW OF THE '985 PATENT	39
VII.	OVERVIEW OF THE PRIOR ART	41
A.	Chalom	41
B.	Ganin	43
C.	Aslan	45
D.	Grau	47
E.	Pace	47
F.	Liu	47
VIII.	CLAIM CONSTRUCTION	49
IX.	ANTICIPATION	50
X.	OBVIOUSNESS	50
XI.	OPINIONS REGARDING PATENTABILITY	51

XII.	MY OPINIONS AS TO THE CHALLENGED CLAIMS OF THE	
	'985 PATENT	53
A.	My Opinions as to Claims 1-3, 5, 10, 12-14, and 18 in view of	
	Ganin alone	53
1.	Claim 1	53
a.	1[P]	55
b.	1[A].....	57
c.	1[B].....	59
d.	1[C].....	60
e.	1[D].....	62
f.	1[E]	71
g.	1[F]	75
h.	1[G].....	78
2.	Claim 2.....	79
3.	Claim 3.....	87
4.	Claim 5.....	89
a.	5[P] and 5[A]	89
b.	5[B].....	91
5.	Claim 10.....	92
6.	Claim 12.....	92
7.	Claim 13.....	93
a.	13[P]	93
b.	13[A].....	94
c.	13[B].....	95
8.	Claim 14.....	96
9.	Claim 18.....	97
a.	18[P] and 18[A-E]	99
b.	18[F]	101

B.	My Opinions as to Claims 16 and 19 based on Ganin in view of Liu.....	102
1.	Claim 16.....	102
a.	16[P]	104
b.	16[A-G]	104
c.	16[H].....	107
2.	Motivation to Combine Ganin with Liu.....	107
3.	Claim 19.....	110
a.	19[P] and 19[A-E]	111
b.	19[F-G]	113
C.	My Opinions as to Claims 1-3, 5-6, 8-14, and 16-19 in view of the Chalom-Aslan Combination.....	113
1.	Motivation to Combine Chalom and Aslan	113
2.	Claim 1.....	125
a.	1[P]	127
b.	1[A].....	134
c.	1[B].....	135
d.	1[C].....	137
e.	1[D-E]	140
f.	1[F-G]	154
3.	Claim 2.....	161
4.	Claim 3.....	164
5.	Claim 5.....	166
6.	Claim 6.....	167
7.	Claim 8.....	170
8.	Claim 9.....	172
9.	Claim 10.....	173
10.	Claim 11.....	173
11.	Claim 12.....	174

12.	Claim 13.....	174
	a. 13[P]	175
	b. 13[A].....	175
	c. 13[B].....	176
13.	Claim 14.....	177
14.	Claim 16.....	179
	a. 16[P]	181
	b. 16[A]-[C]	183
	c. 16[D] and 16[F]	184
	d. 16[E] and 16[G].....	185
	e. 16[H].....	186
15.	Claim 17.....	186
	a. 17[P] and 17[A-E]	188
	b. 17[F]	189
16.	Claim 18.....	191
	a. 18[P] and 18[A-E]	192
	b. 18[F]	194
17.	Claim 19.....	196
	a. 19[P] and 19[A-E]	197
	b. 19[F]	199
	c. 19[G].....	199
D.	My Opinions as to Claim 4 based on Chalom in view of Aslan and Grau	199
	1. Claim 4.....	199
	2. Motivation to Combine Chalom-Aslan with Grau.....	201
E.	My Opinions as to Claim 7 in view of Chalom-Aslan and Pace	207
	1. Claim 7.....	207
	2. Motivation to Combine Chalom-Aslan with Pace	208
XIII.	CONCLUDING STATEMENTS.....	210

EXHIBIT LIST

EXHIBITS FILED BY PETITIONERS	
Ex. 1001	U.S. Patent No. 10,740,985 to Eric Sommerlade and Alexandros Neophytou (“’985 Patent”)
Ex. 1002	Patent Prosecution History of U.S. Patent No. 10,740,985
Ex. 1003	Declaration of L. Karam Regarding Invalidity of U.S. Patent No. 10,740,985
Ex. 1004	Curriculum Vitae of Lina Karam
Ex. 1005	U.S. Patent No. 10,664,949 to Edmond Chalom and Or Shimshi (“Chalom”)
Ex. 1006	Ganin <i>et al.</i> “Deepwarp: Photorealistic image resynthesis for gaze manipulation.” <i>ECCV 2016: 14th European Conference on Computer Vision, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II (pp. 311-326)</i> . Springer International Publishing. (“Ganin”)
Ex. 1007	U.S. Patent Application Publication No. 2017/0132528 A1 to Aslan <i>et al.</i> (“Aslan”)
Ex. 1008	Webpage printout at https://link.springer.com/chapter/10.1007/978-3-319-46475-6_20 (DeepWarp: Photorealistic Image Resynthesis for Gaze Manipulation)

Ex. 1009	U.S. Patent Application Publication No. 2018/0158246 A1 to Oliver Grau and Daniel Pohl (“Grau”)
Ex. 1010	U.S. Patent No. 8,942,283 to Charles P. Pace (“Pace”)
Ex. 1011	U.S. Provisional Application No. 62/542,661 filed August 8, 2017 (“’661 Prov. Appl.”)
Ex. 1012	U.S. Provisional Application No. 62/543,587 filed August 10, 2017 (“’587 Prov. Appl.”)
Ex. 1013	Prosecution of European Patent Application No. 18844651
Ex. 1014	[Reserved]
Ex. 1015	[Reserved]
Ex. 1016	Liu <i>et al.</i> “Optical flow and principal component analysis-based motion detection in outdoor videos.” <i>Eurasip Journal on Advances in Signal Processing</i> , volume 2010, article id 680623, 6 pages, 2010. Hindawi Publishing Corporation. (“Liu”)
Ex. 1017	Webpage printout at https://www.reald.com/privacypolicy (RealD Privacy Policy)
Ex. 1018	Webpage printout at https://web.archive.org/web/20161231162955/https://www.skoltech.ru/en/2016/11/ckolteh-litsenziroval-tehnologiyu-

	obespecheniya-zritelnogo-kontakta-v-videokonferentsiyah-v-kompaniyu-reald/ (Skoltech Licenses Eye Contact Videoconferencing Technology to RealD)
Ex. 1019	Kononenko, Daniil. “Learnable Warping-Based Approach To Image Re-Synthesis With Application to Gaze Redirection.” (2017): 1-131. (Kononenko Doctoral Thesis)
Ex. 1020	Sotera Stipulation
Ex. 1021	Yang <i>et al.</i> “Eye gaze correction with stereovision for video-teleconferencing.” <i>ECCV 2002: 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28–31, 2002, Proceedings, Part II</i> (pp. 479-494). Springer Berlin Heidelberg, 2002. (“Yang”)
Ex. 1022	Cham <i>et al.</i> “Analogous view transfer for gaze correction in video sequences.” <i>7th International Conference on Control, Automation, Robotics and Vision, Singapore, December 2-5, 2002, Proceedings</i> (pp. 1415-1420). IEEE Xplore. (“Cham”)
Ex. 1023	Webpage printout at: https://ieeexplore.ieee.org/abstract/document/1234980 (Analogous view transfer for gaze correction in video sequences)
Ex. 1024	Wood <i>et al.</i> “Gazedirector: Fully articulated eye gaze redirection in video.” <i>arXiv: 1704.08763v1 [cs.CV]</i> , 27 April 2017.
Ex. 1025	Zitnick <i>et al.</i> “Manipulation of video eye gaze and head orientation for video teleconferencing.” <i>Technical Report MSR-TR-99-46, Microsoft Research</i> , 16 June 1999.

Ex. 1026	Webpage printout at: https://www.microsoft.com/en-us/research/publication/manipulation-of-video-eye-gaze-and-head-orientation-for-video-teleconferencing/ (Manipulation of Video Eye Gaze and Head Orientation for Video Teleconferencing – Microsoft Research)
Ex. 1027	Gemmell <i>et al.</i> “Gaze awareness for video-conferencing: A software approach.” <i>IEEE Multimedia</i> , volume 7, issue 4, October-December 2000 (pp. 26-35). (“Gemmell”)
Ex. 1028	Kononenko <i>et al.</i> “Learning to look up: Realtime monocular gaze correction using machine learning.” <i>CVPR 2015: IEEE/CVF Conference on Computer Vision and Pattern Recognition</i> , Boston, MA, USA, 7-12 June 2015 (pp. 4667-4675). (“Kononenko”)
Ex. 1029	Webpage printout at: https://github.com/BlueWinters/DeepWarp (GitHub - BlueWinters_DeepWarp_ Tensorflow implement for DeepWarp)
Ex. 1030	Aleš Hladník, “Image Compression and Face Recognition: Two Image Processing Applications of Principal Component Analysis.” <i>International Circular of Graphic Education and Research</i> , No. 6, January 2013 (pp. 56-61).
Ex. 1031	Complaint: <i>RealD Spark, LLC v. Microsoft Corporation</i> , WDWA-2-22-cv-00942, Dkt. 1.
Ex. 1032	Hinton <i>et al.</i> Distilling the Knowledge in a Neural Network, arXiv:1503.02531v1 [stat.ML], March 2015. (“Hinton”)
Ex. 1033	C. Bucilua, R. Caruana, and A. Niculescu-Mizil. “Model Compression.” <i>KDD 2006: In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge</i>

	<i>Discovery and Data Mining, (KDD), Philadelphia, PA, USA, August 20-23, pages 535–541, 2006 (pp. 535–541). (“Bucilua”)</i>
Ex. 1034	[Reserved]
Ex. 1035	Fleet <i>et al.</i> “Design and Use of Linear Models for Image Motion Analysis.” <i>International Journal of Computer Vision, volume 36, issue 3, 2000 (pp. 171-193).</i> Kluwer Academic Publishers. (“Fleet”)
Ex. 1036	Wulff <i>et al.</i> “Efficient Sparse-to-Dense Optical Flow Estimation using a Learned Basis and Layers.” <i>CVPR 2015: IEEE/CVF Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7-12 June 2015 (pp. 120-130).</i> (“Wulff”)
Ex. 1037	Burt <i>et al.</i> “A Multiresolution Spline With Application to Image Mosaics.” <i>ACM Transactions on Graphics, volume 2, issue 4, October 1983 (pp. 217-236).</i> (“Burt”)
Ex. 1038	Giger <i>et al.</i> “Gaze correction with a single webcam.” <i>ICME 2014: IEEE International Conference on Multimedia and Expo, Chengdu, China, July 14-18, 2014.</i> (“Giger”)
Ex. 1039	Ho <i>et al.</i> “Gaze correction using 3D video processing for videoconferencing.” <i>ChinaSIP 2015: China Summit and International Conference on Signal and Information Processing, Chengdu, China, July 12-15, 2015 (pp. 496-499).</i> (“Ho”)
Ex. 1040	Jain. <i>Fundamentals of Digital Image Processing.</i> Prentice-Hall, Inc., 1989 (excerpted). (“Jain”)

Ex. 1041	Vetterli <i>et al.</i> Wavelets and Subband Coding. Prentice-Hall, Inc., 1995 (excerpted). (“Vetterli”)
Ex. 1042	Dodge and Karam. Introduction to Machine Learning and Deep Learning: A Hands-On Starter’s Guide, 2017; revised 2019. Available online at https://deeplearningtextbook.org/ . (“Dodge”)
Ex. 1043	Goodfellow <i>et al.</i> Deep Learning, MIT Press, 2016. Available online at https://www.deeplearningbook.org/ (excerpted). (“Goodfellow”)
Ex. 1044	Murphy. Machine Learning: A Probabilistic Perspective, MIT Press, 2012. (“Murphy”)
Ex. 1045	Sonka <i>et al.</i> Image Processing, Analysis, and Machine Vision, 4 th edition, Cengage Learning, 2015.
Ex. 1046	U.S. Patent No. 10,043,088 (“Odry”)
Ex. 1047	Microsoft Computer Dictionary, fourth edition, 1999 (excerpted)
Ex. 1048	Internet Archive affidavit w exhibits
Ex. 1049	[Reserved]
Ex. 1050	Webpage printout at: https://link.springer.com/book/10.1007/978-3-319-46475-6 (Webpage Computer Vision – ECCV 2016)

Ex. 1051	Webpage printout at: https://asp-urasipjournals.springeropen.com/articles/10.1155/2010/680623 (SpringerOpen, Liu)
Ex. 1052	Webpage printout at: https://dl.acm.org/doi/abs/10.1155/2010/680623# (ACM, Liu)
Ex. 1053	U.S. Patent No. 9,338,449 to Joel Sole Rojals, Rajan Laxman Joshi, and Marta Karczewicz (“Rojals”)
Ex. 1054	U.S. Patent Application Publication No. 2008/0043833 A1 to Jane Perera and Joseph D. Young (“Perera”)
Ex. 1055	Joint Claim Construction and Prehearing Statement: <i>RealD Spark, LLC v. Microsoft Corporation</i> , WDWA-2-22-cv-00942, Dkt. 59.
Ex. 1056	George Wolberg, <i>Digital Image Warping</i> , IEEE Computer Society Press, 1990.
Ex. 1057	Webpage printout at: https://ieeexplore.ieee.org/document/7299098 (Learning to look up: Realtime monocular gaze correction using machine learning)
Ex. 1058	Webpage printout at: https://ieeexplore.ieee.org/document/6890306 (Gaze correction with a single webcam)

I, Lina Karam, Ph.D., declare as follows:

I. ENGAGEMENT

1. I reside at 9813 E Lofty Point Rd, Scottsdale, AZ 85262, USA.
2. I have been retained by Microsoft Corporation (“Microsoft” or “Petitioner”) in connection with the above-captioned petition for *Inter Partes* Review (“IPR”) of U.S. Patent No. 10,740,985 to Eric Sommerlade and Alexandros Neophytou (“the ’985 Patent,” Ex. 1001). I understand the ’985 Patent is currently assigned to RealD Spark LLC (“Patent Owner”).
3. I have been asked by Petitioner to offer opinions regarding the ’985 Patent, including the unpatentability of claims 1-14 and 16-19 (which I may refer to subsequently as the “challenged claims”) in view of certain prior art. This declaration sets forth the opinions I have reached to date regarding these matters.
4. I am being compensated by Petitioner at my standard hourly consulting rate for my time spent on this matter. My compensation is not contingent on the outcome of the IPR or on the substance of my opinions.
5. I have no financial interest in Petitioner or Patent Owner.

II. PROFESSIONAL BACKGROUND

6. My qualifications are set forth in my curriculum vitae, a copy of which is attached as an Exhibit 1004. As set forth in my curriculum vitae:

7. I received a Bachelor of Engineering (B.E.) from the American University of Beirut in Computer and Communications Engineering in 1989, and a M.S. and a Ph.D. from the Georgia Institute of Technology in Electrical Engineering in 1992 and 1995, respectively.

8. I am currently an Emerita Professor in the School of Electrical, Computer, and Energy Engineering at Arizona State University (ASU) and the Director of the R&D Image, Video, and Usability (IVU) Laboratory at ASU. Prior to becoming Emerita Professor, I was a tenured Full Professor, Computer Engineering Program Chair and Computer Engineering Director for Industry Engagement at ASU. From 2020 to 2022, I served as Dean of the School of Engineering at the Lebanese American University. From 2019 to 2021, I served as the Editor-In-Chief of the high-impact IEEE Journal of Special Topics in Signal Processing. I am also the President of PICARIS, LLC, a consulting company that provides expertise on media processing, compression, computer vision, and machine learning.

9. My research includes topics such as Signal, Image, and Video Processing, Compression, and Transmission; Computer Vision; Machine Learning; Perceptual-based Processing; Visual Attention Models; Automated Quality Assessment and Monitoring; Multidimensional Signal Processing; and Digital Filter Design. I have various awards in recognition of my research, including U.S.

National Science Foundation CAREER Award, Intel Outstanding Researcher Award, U.S. National Aeronautics and Space Administration (NASA) Technical Innovation Award, the IEEE Signal Processing Society (SPS) Best Journal Paper Award, IEEE Phoenix Section Outstanding Faculty Award, and IEEE Region 6 Award. I am featured in the 8th and 9th editions of Research.com (2022 and 2023) among World Top Computer Science Scientists.

10. In addition, I am an Institute of Electrical and Electronics Engineers (IEEE) Fellow, which is the highest grade level conferred by IEEE each year to no more than one-tenth of 1% of all IEEE voting members, for my contributions in the areas of image and video processing, visual media compression and transmission, and digital filtering. In addition to serving as Editor-In-Chief of a prestigious IEEE journal, I have also served on several IEEE Boards and chaired multiple IEEE committees. For example, I served on the IEEE Signal Processing Society (SPS) Board of Governors (2016-2018), IEEE SPS Publications Board (2019-2021), IEEE SPS Award Board (2019-2020), IEEE SPS Conference Board (2003-2005 & 2017-2018), IEEE Publication Services and Products Board (PSPB) Strategic Planning Committee (2014-2016), IEEE Technical Activities Board (TAB)/PSPB Product and Services Committee, IEEE Educational Activities Board (EAB) Faculty Resources Committee, and IEEE TechRxiv Advisory and Editorial Boards. I am also an expert delegate of the ISO/IEC JTC1/SC29 Committee (Coding of

audio, picture, multimedia, and hypermedia information) and participating in JPEG/MPEG standardization activities.

11. As part of my various roles at ASU, PICARIS, ISO and IEEE, I am actively involved in industry. For example, as an intern I contributed to the development of image and video processing and compression at AT&T Bell Labs (Murray Hill), and multi-dimensional data processing and visualization at Schlumberger. Later in my career I collaborated in the research and development of computer vision, machine learning, image/video processing, compression, and transmission projects with various industry leaders, including Intel, Qualcomm, Google, NTT, Motorola, Freescale, NXP, General Dynamics, and NASA.

12. In addition, I directed the development of image/video processing, compression, computer vision and machine learning technologies. The developed computer vision and machine learning systems for automated defect detection, 3D characterization, gender/age prediction, and autonomous vehicles were adopted at Intel. The developed image-based non-wet solder joints detection system was granted a Divisional Recognition Award by Intel. More details can be found in Said, Bennett, Karam, and Pettinato, "Automated Detection and Classification of Non-Wet Solder Joints," IEEE Transactions on Automation Science and Engineering, Jan 2011. The developed image-based automated void detection system helped in enabling two industry standards, JEDEC JC 14-1 void guideline

and IPC-7095C. More details can be found in Said, Bennett, Karam, Siah, Goodman, and Pettinato, “Automated Void Detection in Solder Balls in the Presence of Vias and Other Artifacts,” IEEE Transactions on Electronics Packaging Manufacturing, November 2012. More details about the developed 3D characterization system including image-based solder ball height and warpage measurements can be found in Li, Bennett, Karam, and Pettinato, “Stereo Vision Based Automated Solder Ball Height and Substrate Coplanarity Inspection,” IEEE Transactions on Automation Science and Engineering, April 2016, and in Karam and Li, “Stereo Vision Measurement System and Method,” US Patent 9,704,232 B2, issued July 2017. Details about machine learning based computer vision for defect detection can be found in Haddad, Yang, Karam, Ye, Patel and Braun, “Multi-Feature, Sparse-Based Approach for Defects Detection and Classification in Semiconductor Units,” IEEE Transactions on Automation Science and Engineering, August 2016, and in Haddad, Dodge, Yang, Karam, Ye, Patel and Braun, “Locally Adaptive Statistical Background Modeling with Deep Learning-Based False Positive Rejection for Defect Detection in Semiconductor Units,” IEEE Transactions on Semiconductor Manufacturing, August 2020.

13. The developed machine learning and computer vision systems for autonomous vehicles include Advanced Driver Assist Systems (ADAS) technologies. The developed real-time forward collision warning system prototype

was demonstrated by our Intel industry collaborators at the 2015 Consumer Electronics Show (CES) in Las Vegas. More details can be found in Prakash, Li, Akhbari, and Karam, “Sparse Depth Calculation using Real-time Key-point Detection and Structure from Motion for Advanced Driver Assist Systems,” Springer Lecture Notes, ISVC, December 2014, and in Prakash, Akhbari, and Karam, “Robust Obstacle Detection for Advanced Driver Assistance Systems using Distortions of Inverse Perspective Mapping of a Monocular Camera,” Robotics and Autonomous Systems Journal, 2019.

14. I directed the development of scalable visual compression technologies. The developed codecs, utilizing the scalable visual compression technologies, were commercialized by General Dynamics. More details can be found in Chien, Sadaka, Abousleman, and Karam, “Region-of-Interest-Based Ultra Low-Bit-Rate Video Coding,” SPIE Symposium on Defense & Security, March 2008. I also directed the development of perceptual-based visual compression methods and algorithms. The work on JPEG2000 Encoding with Perceptual Distortion Control enabled the integration of adaptive perceptual-based visual processing and compression in the JPEG 2000 image coding standard and demonstrated improved performance in terms of visual quality and compression, while maintaining full compatibility with the JPEG 2000 standard. For this

perceptual-based image compression work, I received a Technical Innovation Award from NASA.

15. In addition, I directed the development of automated biomedical image processing and computer vision technologies that enable high-throughput cancer diagnostics and drug discovery. The developed automated image analysis technologies were licensed for cancer research at different industries and institutions, including the Translational Genomics Institute (TGEN) and the New York School of Medicine. More details can be found in Karam and Said, “Automatic Cell Migration and Proliferation Analysis,” US Patent 9,082,164, issued July 2015.

16. My work has been published numerous times in prestigious journals, including the IEEE Transactions on Signal Processing, IEEE Transactions on Image Processing, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), IEEE Transactions on Broadcasting, and IEEE/CVF Computer Vision and Pattern Recognition (CVPR) Conference. Some selected publications include “Morphological Reversible Contour Representation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, March 2000, “Morphological Text Extraction from Images,” IEEE Transactions on Image Processing,” Nov. 2000, “A No-Reference Objective Image Sharpness Metric Based on the Notion of Just Noticeable Blur (JNB),” IEEE Transactions on Image

Processing, April 2009, “Automated Detection and Classification of Non-Wet Solder Joints,” IEEE Transactions on Automation Science and Engineering, Jan. 2011, “Objective Video Quality Assessment Methods: A Classification, Review, and Performance Comparison,” IEEE Transactions on Broadcasting, June 2011, “An Efficient Selective Perceptual-Based Super-Resolution Estimator,” IEEE Transactions on Image Processing, Dec. 2011, “Select Trends in Image, Video, and Multidimensional Signal Processing,” IEEE Signal Processing Magazine, Jan. 2012, “A Distributed Canny Edge Detector: Algorithm and FPGA Implementation,” IEEE Transactions on Image Processing, July 2014, “3D Blur Discrimination,” ACM Transactions on Applied Perception, May 2016, “Spatially-Varying Blur Detection Based on Multiscale Fused and Sorted Transform Coefficients of Gradient Magnitudes,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, “Unconstrained Ear Recognition using Deep Neural Networks,” IET Biometrics, January 2018, “Visual Saliency Prediction Using a Mixture of Deep Neural Networks,” IEEE Transactions on Image Processing, August 2018, “Quality Robust Mixtures of Deep Neural Networks,” IEEE Transactions on Image Processing, November 2018, “Augmented Sparse Representation Classifier (ASRC) for Face Recognition under Quality Distortions,” IET Biometrics Journal, November 2019, “DeepCorrect: Correcting DNN Models against Image Distortions,” IEEE

Transactions on Image Processing, December 2019, “It GAN DO Better: GAN-based Detection of Objects on Images with Varying Quality,” IEEE Transactions on Image Processing, November 2021. I also co-authored a machine learning book entitled “*Introduction to Machine Learning and Deep Learning: A Hands-On Starter's Guide*,” published in 2017, revised in 2019, and available online at <https://deeplearningtextbook.org/>.

17. I am an inventor, and I have been awarded eight patents in the field of image and video processing, compression, and transmission: U.S. Patent Nos. 6,154,493 (“The Compression of Color Images Based on a 2-Dimensional Discrete Wavelet Transform Yielding a Perceptually Lossless Image”), 6,124,811 (“Realtime Algorithms and Architectures for Coding Images Compressed by DWT-Based Techniques”), 6,717,990 (“Communication System and Method for Multi-Rate, Channel-Optimized Trellis-Coded Quantization”), 7,551,671 (“System and Method for Transmission of Video Signals using Multiple Channels”), 9,082,164 (“Automatic Cell Migration and Proliferation Analysis”), 9,501,710 (“Systems, Methods, and Media for Identifying Object Characteristics Based on Fixation Points”), 9,704,232 (“Stereo Vision Measurement System and Method”), and 11,0304,85 (“Systems and Methods for Feature Corrections and Regeneration for Robust Sensing, Computer Vision, and Classification,”). I was also a named

inventor on an additional patent application, published as U.S. 2008/0259796

(“Method and Apparatus for Network-Adaptive Video Coding”).

18. Additional information regarding my background, qualifications, publications, and presentations is provided in my CV, which I understand has been submitted as Exhibit 1004.

III. MATERIALS REVIEWED

19. In forming my opinions, I have reviewed the '985 Patent and considered each of the documents listed in the Exhibit List above and all other materials discussed herein. In reaching my opinions, I have relied upon my experience in the field and also considered the viewpoint of a person of ordinary skill in the art as of the time of the earliest claimed priority date of the '985 Patent, *i.e.*, August 8, 2017. As explained below, I am familiar with the level of a person of ordinary skill in the art regarding the technology at issue as of that time and all of the opinions I have formed have been assessed from the perspective of a person of ordinary skill in the art.

IV. DESCRIPTION OF THE RELEVANT TIMEFRAME, THE RELEVANT FIELD, AND A PERSON OF ORDINARY SKILL IN THE ART

20. I understand that the '985 Patent was filed on August 7, 2018, and claims priority to Provisional Application No. 62/542,661 (Ex. 1011), filed on August 8, 2017, and Provisional Application No. 62/543,587 (Ex. 1012), filed on

August 10, 2017. Thus, for purposes of my analysis, I have treated the effective filing date for the challenged claims as August 8, 2017. I reserve the right to update my analysis should Patent Owner assert a different priority date.

21. I have received and understand the specification, claims, and file history of the '985 Patent.

22. In my opinion, a person having ordinary skill in the art ("PHOSITA") at the time of the effective filing date of the '985 Patent would have had at least an undergraduate degree in computer science, electrical engineering, computer engineering, or equivalent education, and at least two years of work experience using machine learning for image processing. Additional education might substitute for less work experience, and vice versa. My opinions presented herein are as viewed through the eyes of a PHOSITA prior to August 8, 2017.

V. TECHNOLOGY BACKGROUND

A. Machine Learning: General

23. A machine learning ("ML") algorithm (or architecture), also referred to as *machine learning model*, is an algorithm (or architecture) that learns from data to perform a task or a class of tasks. Ex. 1043, p. 96. The quality of the learning or task performance is typically assessed using a performance measure and can be optimized using an objective function (or cost function, or loss function) that takes into consideration the desired performance measure and,

possibly, additional constraints that limit the space of solutions based on a-priori knowledge about the task(s) at hand. Ex. 1043, p. 96; Ex. 1042, pp. 27, 30-31.

24. A machine learning model can be non-parametric or parametric. A non-parametric model (e.g., Principal Component Analysis or PCA, Independent Component Analysis or ICA, k-means, k-nearest neighbors or k-NN, etc.) typically makes use of the observed data (i.e., training data) to learn classes or categories without depending on a pre-defined (e.g., fixed size) vector of model parameters. Ex. 1043, pp. 143-147, 481-483, 537; Ex. 1042, p. 24. A parametric machine learning model can be expressed as a parametrized function and takes the general form of $f(\mathbf{x}, \mathbf{w}) = \mathbf{y}$, where $\mathbf{x}$ contains the input data features, $\mathbf{y}$ contains the corresponding class labels or corresponding output values, and $\mathbf{w}$ represents a vector containing the learned (after training) or to be learned (before training) parameters of the machine learning model f . The machine learning model f could take many forms, from traditional machine learning models (e.g., support vector machine, decision tree, random forest, perceptron, shallow neural networks, etc.) to deep learning models with layered deep neural networks (e.g., multi-layer deep neural networks, convolutional neural networks, residual neural networks, recurrent neural networks, etc.) or a combination of these. Ex. 1042, pp. 21-22, 36-38, 49-53; Ex. 1043, pp. 14, 22-24, 96, 137-139, 140-142, 151, 162-163, 363, 480-483, 493-494, 537. $\mathbf{x}$ can take the form of a vector of features. The features

could be some measurements of the data, such as image pixels, or features that are extracted from the image pixels. The label y could be a category label, a continuous valued number, or even a vector of numbers. The model parameters w can be learned based on the given training data (also called training dataset), and based on an optimization algorithm that is used to optimize a cost function (also called objective function). After this optimization procedure, the model is said to be “trained.” The trained model can then be tested or deployed during run-time using a new dataset (referred to as “testing” dataset) that is different from the training dataset that was used to train the model. Ex. 1042, pp. 21-22.

B. Supervised Learning vs Unsupervised Learning

25. Machine learning algorithms can generally be categorized into at least two categories such as supervised learning and unsupervised learning.

26. In supervised machine learning, the model is trained using a training dataset containing labeled data, i.e., the ground-truth label or target is provided for each input data sample. In other words, each training data sample takes the form of a pair $(\mathbf{x}_{\text{train}}, \mathbf{y}_{\text{train}})$, where $\mathbf{x}_{\text{train}}$ contains the input data features, and $\mathbf{y}_{\text{train}}$ contains the corresponding ground-truth class label or target value to be predicted by the machine learning model. Ex. 1043, p. 102; Ex. 1044, p. 2. The '985 Patent, Ganin, and Chalom all involve supervised machine learning.

27. In unsupervised machine learning, the model is trained using a training dataset containing unlabeled data, i.e., only the input data samples $\mathbf{x}_{\text{train}}$ are provided. The model learns by discovering patterns in the data, relationships between the data samples, and properties (e.g., statistical properties) of the data. Ex. 1043, p. 102; Ex. 1044, 2.

C. Objective Function, Training, Testing, and Deployment

28. Training a machine learning model typically requires a training dataset, objective function (also called cost or loss function), and an optimization algorithm. Optimization refers to the process of either minimizing or maximizing the objective function. When the optimization is formulated as a minimization of the objective function, we can also refer to the objective function as cost function, loss function, or error function. Ex. 1043, p. 79. A popular objective function used in machine learning is the cross-entropy loss function. Ex. 1043, p. 73; Ex. 1042, pp. 27, 43. Another popular objective function is the ℓ_2 -distance also called ℓ_2 -loss discussed in Ganin, which can correspond to the ℓ_2 -norm of an error term or difference term between the output generated by the machine learning model and its corresponding ground-truth. Ex. 1006, p. 8.

29. For a parametric machine learning model, the objective function is expressed in terms of the model parameters $\mathbf{w}$ and these model parameters are learned by optimizing the objective function (i.e., by minimizing the cost function).

Popular optimization algorithms include iterative algorithms, such as gradient descent and stochastic gradient descent algorithms, that update the values of the parameters $\mathbf{w}$ at each iteration with the aim to decrease the cost function in an incremental fashion until a desired performance based on a performance measure (e.g., error rate, accuracy, training and generalization errors) is reached. For neural networks, the backpropagation algorithm is additionally used to speed-up the gradient computation and the optimization process. Ex. 1043, pp. 14, 79-83, 147, 197-198; Ex. 1042, pp. 28-29, 36-39.

30. When training a machine learning model, it is important not only to achieve a small error on the training dataset (i.e., small training error) but also to ensure that the trained model can generalize to unseen data samples (i.e., data samples not included in the training dataset; these form the validation or testing dataset). The generalization of the model is tested by deploying the trained model on a validation or testing dataset and checking that the testing error (also known as generalization error) is small enough by checking that the gap between the training and testing error is small (i.e., the average testing error value is close to the average training error value). Having a small training error but a large testing error corresponds to the model overfitting the training data and not being able to generalize to unseen data samples. A large training error corresponds to the model underfitting the training data. Both overfitting and underfitting should be avoided.

This can be done by running the training process through more iterations or restarting the training using different initializations. Tuning the model hyperparameters (i.e., settings or parameters that are not learned but that are set or adjusted to control the algorithm's behavior) is typically performed when either restarting or continuing the training process in order to achieve the desired performance. Ex. 1043, pp. 107-118; Ex. 1042, pp. 22-23.

31. Once the training is completed, the resulting trained model can be deployed during run-time to perform the targeted task on new input data samples that are obtained from a testing set or from the real-world. For a parametric machine learning model, training the model corresponds to determining the parameters $\mathbf{w}$ of the function $f(\mathbf{x}, \mathbf{w}) = \mathbf{y}$, and consequently it corresponds to estimating the function $f(\mathbf{x}, \mathbf{w})$. Given a new input data sample $\mathbf{x}_{\text{test}}$, the function can predict the value $\mathbf{y}_{\text{test}}$ corresponding to $\mathbf{x}_{\text{test}}$.

D. Common Machine Learning Algorithms and Architectures

32. An overview of select machine learning models, including neural networks, random forests, and teacher-student architectures, is provided below.

1. *Neural Networks*

33. Neural networks (“NNs”) are parametric machine learning models that are motivated by biological learning. They consist of distributed elementary computational units called “neurons” that can be stacked together within the same

layer and across different layers. A neuron can be represented by a parametric function $y = f(\mathbf{x}, \mathbf{w}) = f_{\mathbf{w}}(\mathbf{x})$ that takes the form of a linear function followed by a non-linearity:

$$y = f_{\mathbf{w}}(\mathbf{x}) = \rho(\mathbf{x} \cdot \mathbf{w}^T) = \rho(w_1x_1 + \cdots + w_nx_n)$$

where $\mathbf{x} = (x_1, \cdots, x_n)$ and $\mathbf{w} = (w_1, \cdots, w_n)$ are two row vectors of length n representing, respectively, the input data features and model parameters for the considered neuron, $\rho()$ is a non-linear function (e.g., sigmoid function or rectified linear unit (ReLU); *see* Ex. 1042, pp. 42-43; Ex. 1043, p. 169, Fig. 6.3), and y is the output scalar value of the neuron. The T superscript corresponds to the transpose operation applied to the row vector $\mathbf{w}$, transforming it into a column vector. Typically, the linear portion $\mathbf{x} \cdot \mathbf{w}^T$ is generalized to include a scalar bias term (corresponding to an additional model parameter) while still being referred to as linear:

$$y = f_{\mathbf{w}}(\mathbf{x}) = \rho(\mathbf{x} \cdot \mathbf{w}^T + b) = \rho(w_1x_1 + \cdots + w_nx_n + b)$$

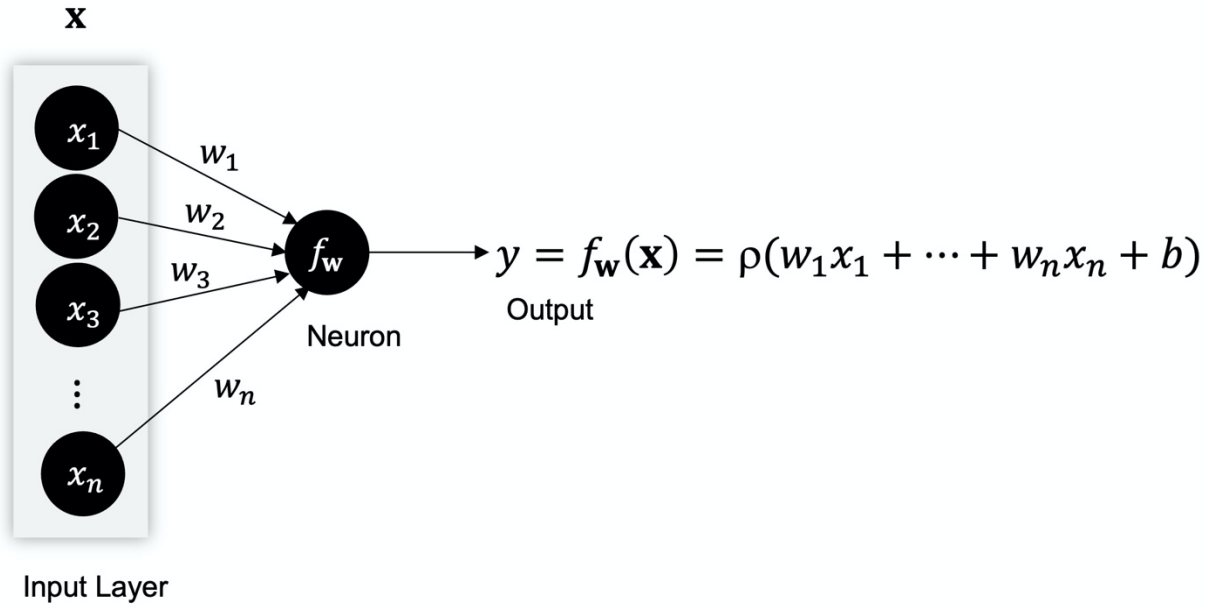


Figure 1. A single neuron and its output.

34. “Neurons” within the same layer can operate in parallel. The more neurons within the same layer the wider the neural network becomes. Layers of neurons within a neural network are cascaded one after each other, with the output of a previous layer becoming the input of the next layer. The more layers in a network the deeper the network becomes. A deeper network can perform significantly better than a wider one (Ex. 1043, pp. 196-197, Figs. 6.6-6.7). In deep learning, deep neural networks are used.

35. Consider a neural network with L cascaded layers. A neural network’s layer l ($l = 1, \dots, L$) with m neurons can be represented by a multi-input, multi-output parametric function:

$$\mathbf{y} = f_{\mathbf{w}_l}(\mathbf{x}) = \rho(\mathbf{x} \cdot \mathbf{W}_l^T + \mathbf{b}_l)$$

where $\mathbf{x} = (x_1, \dots, x_n)$ and $\mathbf{y} = (y_1, \dots, y_m)$ are two row vectors representing, respectively, the input data features and the output data features of the layer l (each neuron generates one output data value; so, the m neurons generate m output data values); $\mathbf{W}_l$ and $\mathbf{b}_l$ store the model weight and bias parameters, respectively, with $\mathbf{W}_l$ being a matrix of size $m \times n$ and $\mathbf{b}_l$ being an m -length row vector:

$$\mathbf{W}_l = \begin{bmatrix} \mathbf{w}_{l,1} \\ \vdots \\ \mathbf{w}_{l,m} \end{bmatrix} = \begin{bmatrix} w_{l,1,1} & \dots & w_{l,1,n} \\ \vdots & \ddots & \vdots \\ w_{l,m,1} & \dots & w_{l,m,n} \end{bmatrix}$$

$$\mathbf{b}_l = [b_{l,1} \quad \dots \quad b_{l,m}]$$

It should be noted that neuron i ($i=1, \dots, m$) in layer l has its learnable parameters stored in weight vector $\mathbf{w}_{l,i}$ (i.e., i^{th} row of weight matrix $\mathbf{W}_l$) and in bias value $b_{l,i}$ (i.e., i^{th} element of bias vector $\mathbf{b}_l$).

36. The neural network model with L cascaded layers can thus be represented by a parametric function that is the composition of the layers' functions:

$$\mathbf{y} = f_{\mathbf{w}}(\mathbf{x}) = f_{\mathbf{W}_L} \circ f_{\mathbf{W}_{L-1}} \circ \dots \circ f_{\mathbf{W}_2} \circ f_{\mathbf{W}_1}(\mathbf{x}) = f_{\mathbf{W}_L} \left(f_{\mathbf{W}_{L-1}} \left(\dots f_{\mathbf{W}_2} \left(f_{\mathbf{W}_1}(\mathbf{x}) \right) \right) \right)$$

37. The network's input $\mathbf{x}$ corresponds to the input layer of the network and the network's last layer corresponds to the output layer of the network; the other layers are referred to as *hidden* layers and correspond to feature extractors or feature generators. Ex. 1044, p. 566; Fig. 16.11; Ex. 1043, p. 169, Fig. 6.2.

38. A layer in a neural network is said to be a fully connected layer when each neuron in that layer is connected to all the layer's input data; in other words, each neuron receives as input and operates on all the layer's input data. Ex. 1042, p. 40. A layer in a neural network is said to be non-fully-connected when at least one of its neurons operates only on a subset of its input. The region containing the input samples on which a neuron operates (or to which a neuron is connected) is called the receptive field of the neuron. For a fully connected layer, the receptive field of each neuron in the layer consist of all the input data; for a non-fully connected layer, the receptive field is limited to a subset of the input data for one or more neurons in that layer, i.e., some of the weight parameter values associated with that neurons are set to zero. Ex. 1043, p. 327, Fig. 9.4.

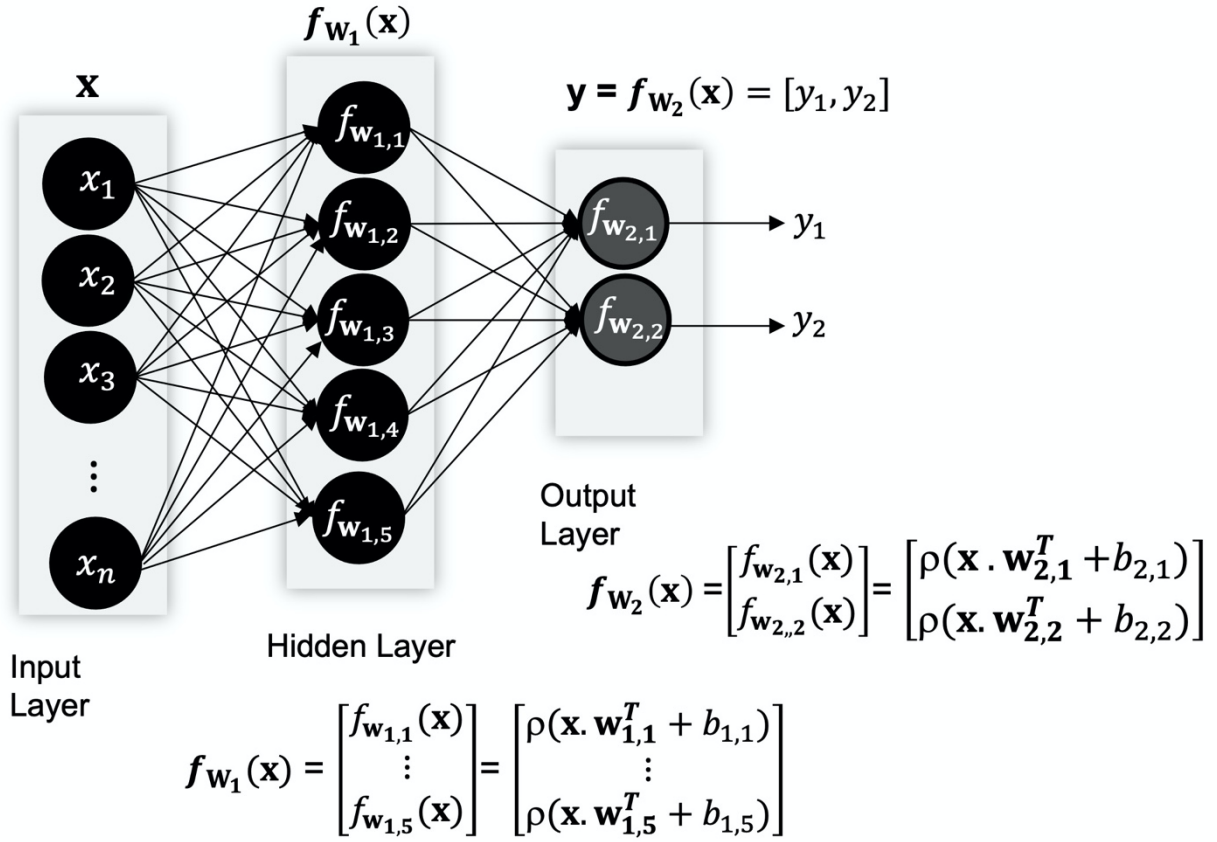


Figure 2. A fully-connected neural network with a single hidden layer.

39. A *convolutional* layer in a neural network is a non-fully-connected layer whose neurons share the same weights, and whose receptive fields have the same size but cover different (can be overlapping) locations of the input data. For a convolutional layer, the columns in the weight matrix $\mathbf{W}_l$ are typically sparse (i.e., mostly zeros) with only few non-zero values; furthermore, the columns are shifted versions of each other with the non-zero values being equal in all columns but their location being shifted. In this case, the matrix multiplication operation

$\mathbf{x} \cdot \mathbf{W}_l^T$ can be efficiently implemented using a convolution operation. Ex. 1043, pp. 321-330, Figs. 9.2, 9.3, 9.4, 9.5; Ex. 1044, pp. 566-567, Fig. 16.14; Ex. 1042, pp. 49-53.

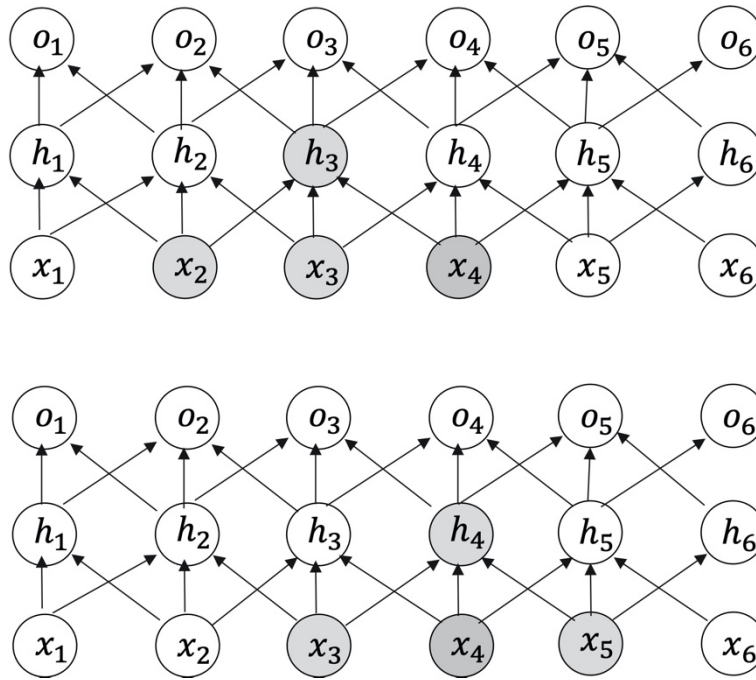


Figure 3. Convolutional neural network. The shaded nodes in the input layer represent the receptive field of neurons in the first hidden layer. Each neuron is only connected to a subset of the input data elements.

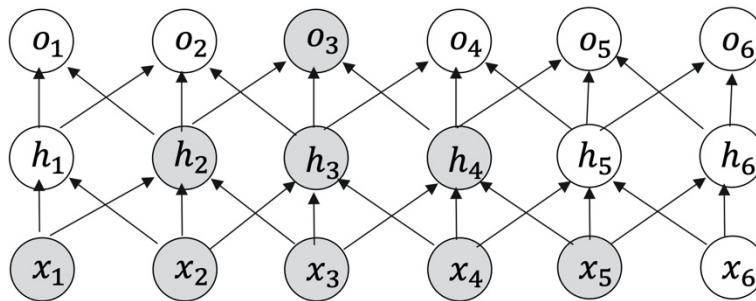


Figure 4. Convolutional neural network (CNN). Each neuron is only connected to a subset of the output of the previous layer. The shaded nodes in the input layer (i.e., first layer with input data) represent the receptive field of neuron o₃. The receptive field of a neuron in the higher (“deeper”) layers of a CNN is larger than the receptive field of a neuron in the lower (“shallower”) layers.

40. Additional non-parametric layers can be inserted between the layers' parametric functions such as pooling, subsampling/upsampling, and/or normalization layers. In many applications, it is common to have the final output layer be a non-parametric conversion/normalization layer. Ex. 1043, pp. 176-179, 330-334; Ex. 1044, pp. 568-569, Fig. 16.14. For example, in multi-class classification applications, the output layer can be a *softmax* function that converts the previous layer's output values to normalized probabilities. For binary classification applications, the output layer can be a *sigmoid* function. Ex. 1043, pp. 176-179; Ex. 1042, pp. 42-43, 55.

41. A neural network can be a feedforward or recurrent neural network. In a feedforward network, the information flows from the input layer to the output layer forward through the hidden layer with no feedback connections. A feedback connection is a connection in which outputs of the model are fed back as input to the model. Ex. 1042, pp. 75-77, Figs. 6.1-6.2; Ex. 1043, pp. 163, 363, 366-371, Figs. 10.3-10.5.

42. Training a neural network can be performed as described in Sections V.A-V.C for learning the model parameters (elements of $\mathbf{W}$ and $\mathbf{b}$). Once the model parameters are learned, the neural network model can be deployed as indicated in Section V.C.

2. *Random Forests*

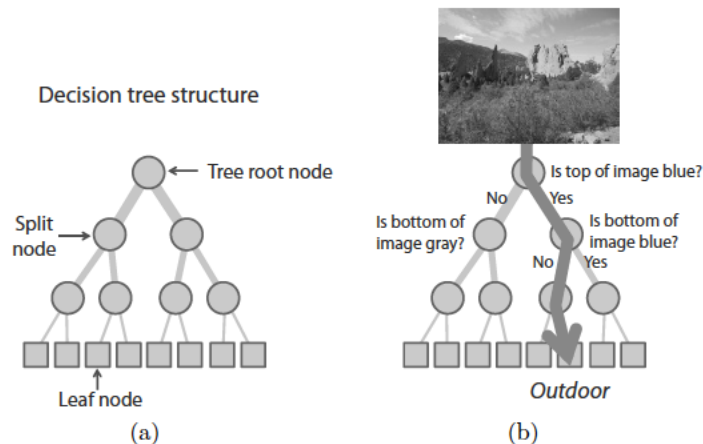
43. A random forest consists of an ensemble of decision trees (e.g., M decision trees). The M decision trees are each trained on different subsets of the training data; these subsets are formed by randomly sampling the training data with replacement. Ex. 1044, pp. 552-553. The output of the random forest model is obtained by combining (e.g., by averaging) the outputs, also known as predictions or decisions, of its individual trained decision trees to obtain a single prediction:

$$f_{RF}(\mathbf{x}) = \frac{1}{M} \sum_{t=1}^M f_t(\mathbf{x})$$

where $\mathbf{x} = (x_1, \dots, x_n)$ consists of the input data features, $f_t(\mathbf{x})$ is the output of the t^{th} decision tree, and $f_{RF}(\mathbf{x})$ is the output of the random forest. Such technique for building an ensemble model is called bagging. Ex. 1044, pp. 552-553; Ex. 1045, pp. 443, 445, 447, Figs. 9.36, 9.37, 9.39-9.40.

44. These decision trees are also known as the base learners of the random forest. By combining the outputs of its decision trees, the random forest can result in a higher prediction accuracy as compared to the output of a single decision tree; it is also able to reduce overfitting and thus can provide better generalization to unseen data (i.e., testing data or data that is not part of the training dataset). Ex. 1044, pp. 552-553; Ex. 1045, p. 448.

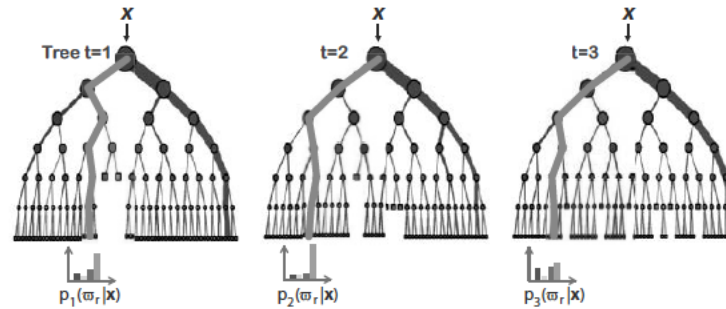
45. Each decision tree is a machine learning model that can be represented by a tree graph consisting of a root node, internal (also called split) nodes and terminal leaf nodes. The nodes are connected via branches across the tree levels. A *parent* node connects to other nodes (known as its *children* nodes) at a subsequent level of the tree. A leaf node does not have any children nodes. The total number of levels in a tree is the tree depth. Each decision tree can be deployed to make decisions for classification tasks (e.g., predicting a class label or category) and/or regression tasks (e.g., predicting a continuous numeric value).



Ex. 1045, p. 443, Fig. 9.35.

46. Each decision tree with L leaves partitions the input data feature space into L regions, with each leaf corresponding to one of these regions (or partitions). Each leaf also stores a value (in case of regression tasks) or a class label (in case of a classification task) that is associated with its corresponding region; this stored value or class label is output when the leaf is reached. Instead of a single value or a

single class label, the leaf can store a distribution (or likelihood map), which is more typical for machine learning applications since it is less common for a leaf to be “pure” (i.e., to have only data samples corresponding to a single value or a single class reaching that leaf). The distribution gives the likelihood or probability of a value or class label when the leaf is reached. For example, for a classification task with R classes, each leaf of a tree t stores a class predictor model that takes the form of an empirical distribution over the R classes, $p_t(\omega_r|\mathbf{x})$, $r \in \{1, \dots, R\}$ (a different one is stored for each leaf of the tree). The empirical distribution $p_t(\omega_r|\mathbf{x})$ is obtained during training from the training data subset that has reached that particular leaf by, for example, counting how many of these data samples belong to each class; the counts can be expressed as percentages to obtain normalized probabilities. Alternatively, the distribution $p_t(\omega_r|\mathbf{x})$ can be obtained during training if samples reaching a leaf have each a corresponding pre-computed distribution (such as an optical flow likelihood map); the final distribution at a leaf is obtained by aggregating (e.g., averaging) all the distributions associated with the data samples that reached that leaf.



Ex. 1045, p. 447, Fig. 9.39.

47. During the training stage of a decision tree, a randomly selected training data subset is used to learn “split” parameters including which subset of input features to split on at each node and, for each selected numerical feature x_i , the corresponding threshold value t_{x_i} (i.e., at a parent node, test if $x_i \leq t_{x_i}$ and, if true, go to left child node; otherwise, if not true, go to right child node). These split parameters are learned by minimizing a cost function (e.g., entropy, misclassification rate, expected error rate/Gini index) or by maximizing a gain function (e.g., information gain, degree of “purity”/homogeneity within space partitions with the final partitions represented by the tree leaves). Ex. 1044, pp. 548-550; Ex. 1045, pp. 444-446.

48. The growth of a decision tree can be stopped by checking whether a node is worth splitting using heuristically determined stopping criteria. Examples of these stopping criteria include: 1) reaching a set maximum tree depth, 2) reaching a sufficiently high degree of homogeneity or purity at children nodes

(e.g., are class labels the same resulting in a pure distribution at right and left children), 3) obtaining an insignificant cost reduction if node is split, 4) obtaining a very small number of samples at either of the children nodes if parent node is split. Ex. 1044, p. 548; Ex. 1045, p. 446.

49. Once the individual decision trees are trained, decision making can be performed, and the random forest model can be deployed. In a random forest, the same input data sample arrives at the root of all the M trees forming the forest. At each node at which the sample arrives, a test optimized during training is employed to direct the sample to one of the two children nodes (i.e., left child node or right child node). Since each tree in the forest is trained using a randomly selected subset of the training data, different split functions (including choice of input features to split on and associated threshold for each node) are learned for each tree; thus, for each tree, the sample can follow a different path until a leaf node is reached. Thus, the random forest produces M outputs, one from each tree in the forest. The output of the random forest is obtained by, e.g., aggregating the M outputs to obtain a single output. For example, when the output stored at each leaf is a distribution over R classes (see figure on previous page), the random forest prediction (output) can be obtained by averaging all tree predictions:

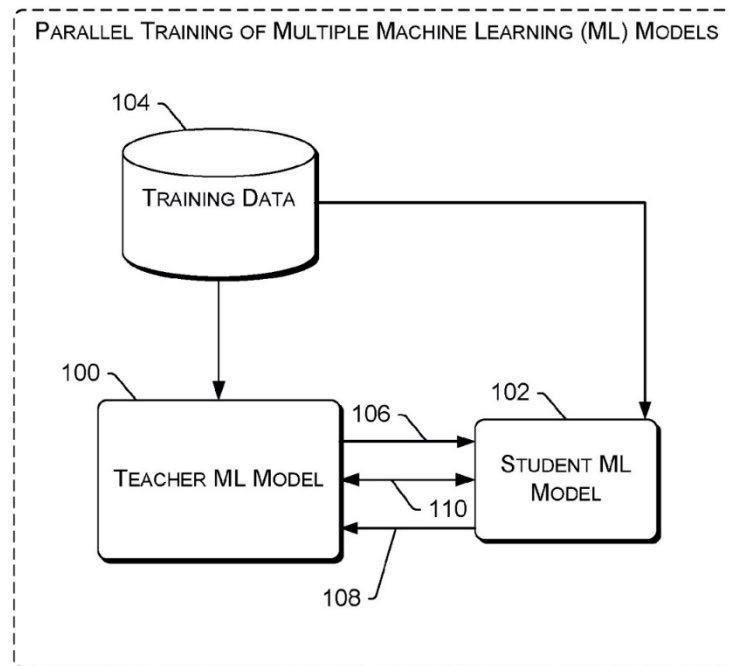
$$p(\omega_r|\mathbf{x}) = \frac{1}{M} \sum_{t=1}^M p_t(\omega_r|\mathbf{x})$$

Ex. 1045, pp. 446-448. The selected value or class is the most probable one, i.e., the one with the highest probability of occurrence based on $p(\omega_r|\mathbf{x})$.

50. Random forests were also used for optical flow prediction and image warping applications. Ex. 1005, 4:19-5:2, 13: 41-65, Fig. 3, Fig. 8, Figs. 12-15; Ex. 1028, pp. 2-3; Ex. 1057. For these applications, the distribution stored at each leaf can take the form of a motion vector (optical flow) likelihood map, where each entry in the likelihood map corresponds to the likelihood of an associated motion vector offset; for a two-dimensional motion vector, an offset consists of a pair of values (mv_x, mv_y) corresponding to a horizontal offset and vertical offset, respectively. Ex. 1005, 12:53-13:7. It should be noted that, for a discrete set of offsets, each possible offset can be considered a class (e.g., the (0,0) offset can be the no motion class or the (0,0) offset class).

3. *Teacher-Student Architecture*

51. In a Teacher-Student architecture, a first pre-trained machine learning model (*Teacher*) is used to help train a second machine learning model (*Student*) through knowledge transfer from the first machine model to the second machine learning model. Ex. 1032, p. 1; Ex. 1007, ¶¶ 0004, 0006, 0021, 0023, Fig. 1.



Ex. 1007, Fig. 1.

52. The teacher machine learning (ML) model is typically more complex or much larger as compared to the student ML model. As such, although the teacher model can achieve a high accuracy, it may not meet desired time and/or storage requirements for real-time deployment due to being too slow or too large.

Ex. 1033, p. 1; Ex. 1032, p. 1; Ex. 1007, ¶ 0045. Once the teacher ML model has been trained or has undergone an initial training phase (e.g., teacher model may subsequently go through additional training phases if needed; *see* Ex. 1007, ¶ 0045), the student model, which is typically simpler or smaller and more suitable for deployment, can be trained by transferring the knowledge from the teacher ML model. For example, using the same training data set as the one used by the teacher

ML model or an expanded one by adding unlabeled data that can be labeled by the teacher model, the student ML model can be trained so that its output can match the output of the teacher model or so that it can match a reference data that is produced by the teacher model and that is passed to the student model. Ex. 1032, p. 2; Ex. 1007, ¶¶ 0034, 0040-0041. Such training can be done by having a term in the objective function (e.g., cost function to be minimized) to reduce the dissimilarity between the teacher and student ML models' outputs. For example, such a loss term can take the form of a cross-entropy loss term or a Bregman divergence distance function that measures the divergence between the outputs of the teacher and student ML models (e.g., soft outputs or set of probabilities). Ex. 1032, pp. 2-3; Ex. 1007, ¶¶ 0032-0038. Alternatively, such a loss term can take the form of a squared difference (e.g., ℓ_2 -distance) between outputs produced by the two models (e.g., hard outputs such as values or labels, or soft outputs such as logits). Ex. 1032, p. 2; Ex. 1007, ¶ 0037. It should be noted that outputs of layers other than the last layers can also be used in combination or in lieu of the last layer's output as part of the knowledge transfer from the teacher ML model for training the student ML model. Ex. 1032, p. 2; Ex. 1007, ¶ 0034.

E. Gaze Correction

53. Gaze correction in video conferencing has been a hot research topic since the early 1990s. Because the camera and the display screen cannot be

physically aligned in a typical computer environment, the person in front of the camera looking at the display screen appears looking away from the camera and thus appears unnatural and lacks eye-contact with the other person.

54. Two main types of gaze correction methodologies have been developed. The first type aims to solve the gaze problem in video conferencing via “synthesizing 3D rotated views of either the entire scene or of the face (that is subsequently blended into the unrotated head).” Ex. 1006, p. 4 (citing Criminisi, A., Shotton, J., Blake, A., Torr, P.H.: Gaze manipulation for one-to-one teleconferencing. In: ICCV (2003); Okada, K.I., Maeda, F., Ichikawaa, Y., Matsushita, Y.: Multiparty videoconferencing at virtual social distance: Majic design. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work. Pp. 385–393. CSCW '94 (1994); Yang, R., Zhang, Z.: Eye gaze correction with stereovision for video-teleconferencing. In: ECCV (2002); Kuster, C., Popa, T., Bazin, J.C., Gostman, C., Gross, M.: Gaze correction for home video conferencing. In: SIGGRAPH Asia (2012); Giger, D., Bazin, J.C., Kuster, C., Popa, T., Gross, M.: Gaze correction with a single webcam. In: IEEE International Conference on Multimedia & Expo (2014)). The other type operates in 2D, where the eye regions, or the pixels thereof, are replaced or modified, such that the resulting eye regions appear to look directly at the camera. *See, e.g.,* Ex. 1028. The '985 Patent, Chalom, and Ganin are all within this type.

55. While the '985 Patent, Chalom, and Ganin all use machine learning in gaze correction, that is not new. For example, “Analogous view transfer for gaze correction in video sequences” is a paper published in 2002, and it uses machine learning for gaze correction. Ex. 1022, Fig. 2, Section 4; Ex. 1023.

56. Moreover, Microsoft has long been an active participant and pioneer in gaze correction research. For example, as early as 1999, Microsoft published a technical report titled “Manipulation of video eye gaze and head orientation for video teleconferencing,” which is authored by C. Lawrence Zitnick, Jim Gemmell, and Kentaro Toyama. Ex. 1025; Ex. 1026. Later in 2000, the same people, plus Thomas Kang from Carnegie Mellon University and Steven Seitz from the University of Washington, published another article, titled “Gaze Awareness for Video-conferencing: A Software Approach.” Ex. 1027.

57. In 2002, Ruigang Yang (while working as a summer intern at Microsoft Research), together with Zhengyou Zhang, published a paper titled “Eye Gaze Correction with Stereovision for Video-Teleconferencing.” Ex. 1021, Title, Fig. 1, fn. 1. The paper recognized the need for eye gaze correction in video conferencing and proposes “a novel approach that is based on stereo analysis combined with rich domain knowledge (a personalized face model).” Ex. 1021, Abstract. As another example, in 2017, Erroll Wood from Microsoft co-authored “GazeDirector: Fully Articulated Eye Gaze Redirection in Video,” which “fit[s] a

multi-part eye region model to video frames using analysis-by-synthesis” and “then redirects gaze by 1) warping the eyelids from the original image using a model-derived flow field, and 2) rendering and compositing synthesized 3D eyeballs onto the output image in a photorealistic manner.” Ex. 1024, Abstract.

F. Optical Flow and Warping

58. Image warping is a branch of the image processing field and relates to the transformation (e.g., geometric or spatial transformation) of images using “an operation that redefines the spatial relationship between points in an image.” Ex. 1056, p. 1; Ex. 1006, p. 3.

59. Applications of image warping include but are not limited to “image registration” (e.g., image alignment), geometric correction (e.g., spatially transforming the image to correct one or more portions in the image), inducing geometric distortions (e.g., geometric variations or spatial deformations induced in input image) for special effects, image enhancement, image editing, and/or image resynthesis. Ex. 1056, pp. 2-4; Ex. 1006, pp. 1-4.

60. Such transformations for image warping include but are not limited to “coordinate transformation” (e.g., pixel displacement) “accounting for translation, scale, rotation, skew, and pincushion effects.” Ex. 1056, p. 2. “For more local control, affine transformations and piecewise polynomial mapping function are widely used, with transformation parameters varying from one region to another.”

Id.; Ex. 1006, pp. 2-4 (“warping field”, “pixel-flow field”, Eq. 2 showing “sampling procedure” implementing local “piecewise” polynomial mapping based on “bilinear interpolation” according to the “flow field” or “warping field”). *See also* Ex. 1035, Abstract, p. 172.

61. In particular, translational coordinate transformation models (“translational models”) are included as part of the affine transformations and “have been used successfully for estimating and representing the optical flow field [e.g., pixel displacement, flow field, warping field]” and they offer a computational advantage by only “requir[ing] two basis flow fields, encoding horizontal and vertical translation, while [more general] affine models require six basis flow fields.” Ex. 1035, p. 172. “These models have been applied locally within small image regions, and globally.” *Id.*

62. Let I be an input image (or image patch) and let O be the output image (or image patch) obtained by warping I , where the warping is performed through local translational transformations according to an optical flow field (also called pixel-flow field, displacement vector field, motion vector field, or warping field). The optical flow field consists of two dimensions, also known as two channels, (u, v) , with the first dimension u corresponding to a horizontal displacement and the second dimension v corresponding to a vertical displacement. Ex. 1035, p.172. Such an optical flow field provides an optical flow vector $(u(x, y), v(x, y))$ at each

pixel location (x, y) in the image or at each pixel location (x, y) in a select set of pixel locations or in a region of interest. The output image O can be obtained by sampling the input image I according to the optical flow vectors $(u(x, y), v(x, y))$ as follows:

$$O(x, y) = I(x + u(x, y), y + v(x, y))$$

Ex. 1028, Eq. 1, p. 3 (of pdf file); Ex. 1016, p. 2. “In other words, the pixel value at (x, y) [in the output image O] is “copy-pasted” from another location [in the input image I] determined by the (u, v) flow vector (u, v) [which corresponds to a two-dimensional (2D) offset or displacement from (x, y) to get to the location of the pixel in I to be copied].” Ex. 1028, p. 3 (of pdf file). Alternatively, the “copy-paste” operation can be used in combination with other mapping and/or image processing operations (e.g., bilinear interpolation, lowpass filtering, texture mapping). Ex. 1006, p. 5, Eq. 2; Ex. 1056, p. 3, p. 5.

63. The optical flow field (i.e., displacement vector field, pixel-flow field, flow field, or warping field) $(u(x, y), v(x, y))$ can be determined using machine learning. Ex. 1006; Ex. 1028; Ex. 1005. During a training phase, the machine learning system or algorithm is trained to predict the displacement vectors $(u(x, y), v(x, y))$ at positions (x, y) in an image region using training data consisting of pairs of input image patches (to which warping is to be applied) and corresponding target ground-truth output image patches (providing the desired

output that needs to be obtained by warping the input). Once trained, the machine learning system or algorithm can be applied during runtime to an input image patch and can provide a prediction of the displacement vector field $(u(x, y), v(x, y))$. The predicted $u(x, y), v(x, y)$ can then be used to warp the input image patch in order to obtain an output image patch with a desired adjustment. Ex. 1006; Ex. 1028; Ex. 1005.

64. For example, the displacement vectors $(u(x, y), v(x, y))$ can be learned (during training) and determined (during runtime) for each pixel (x, y) “based on the appearance of the patch surrounding the pixel” and/or “the location of the pixel [with respect to] feature points” (e.g., anchor points in the image). Ex. 1028, p. 3 (of pdf file). During training, the displacement vectors $(u(x, y), v(x, y))$ can be learned using a matching procedure that “matches each pixel (x, y) to certain pixels within the training data.” Ex. 1028, p. 3 (of pdf file). For example, a block matching procedure can be used wherein “for each pixel location of [input image regions or patches], multiple candidate motion vectors may be evaluated by determining a sum of absolute differences [(SAD)] between a window around the pixel [(i.e., patch surrounding the pixel or pixel neighborhood)] in [input image region or patch] and a motion vector shifted window in target [image] region [(i.e., target output image region or patch)]. For example, the window or evaluation block may have a size of 6x6 pixels over a search range of

10x10 pixels at fixed intervals such as every pixel. Although discussed with respect to 6x6 pixel windows searched over a search range of 10x10 pixels at every pixel, any pixel window size, search range, and interval may be used.” Ex. 1005, 12:23-38. Alternatively, in the aforementioned block matching procedure, the input and output images can be swapped with the multiple candidate motion vectors being evaluated by determining a SAD between a window around the pixel in target output image (or target output patch) and a motion vector shifted window in input image (or input patch), as I show in figure below.

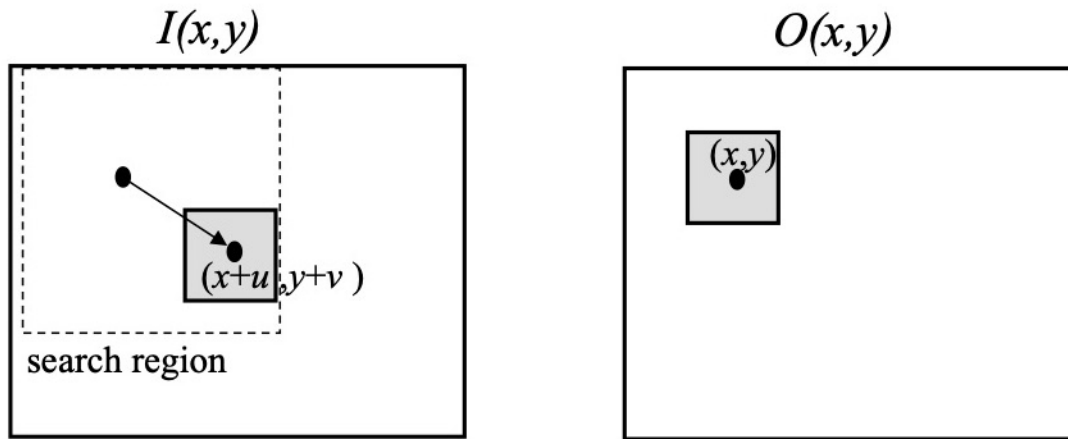


Figure 5. Block matching procedure for optical flow.

65. The matching procedure can be used to generate “likelihood maps that provide candidate motion vectors or inverse probabilities. For example, lower sum of absolute differences [(SAD)] may correlate to higher probability the optimal

motion vector of the candidate motion vectors has been found. For example, the lower the sum of absolute differences at a specific shift (e.g., as provided by the corresponding motion vector), the closer or more likely the true best motion vector corresponds to that shift.” Ex. 1005, 12:39-52, Fig. 12.

VI. OVERVIEW OF THE '985 PATENT

66. I have reviewed the '985 Patent and understand that Eric Sommerlade and Alexandros Neophytou are named as the inventors on this patent. The '985 Patent relates to using machine learning algorithms for “adjusting target features of the digital representation of the head region, for example to correct a perceived gaze direction of eyes.” Ex. 1001, 1:9-12. “The present disclosure is particularly concerned with implementing such processing techniques **with minimal demands on computer hardware and/or power** such that they provide results at or near input data frame rate or user feedback requirements.” *Id.*, 1:50-55.

67. To achieve this, although the '985 Patent does not use the term explicitly, it essentially adopts the well-known “teacher-student” architecture for training a well-known machine learning based gaze correction algorithm. As shown below in Figure 8, to which I added coloring to facilitate understanding, the first machine learning algorithm (i.e., the teacher, green) generates first reference data and the second machine learning algorithm (i.e., the student, blue) learns from

the teacher by using the same training data (purple) and the first reference data (orange) generated by the teacher for the student's training.

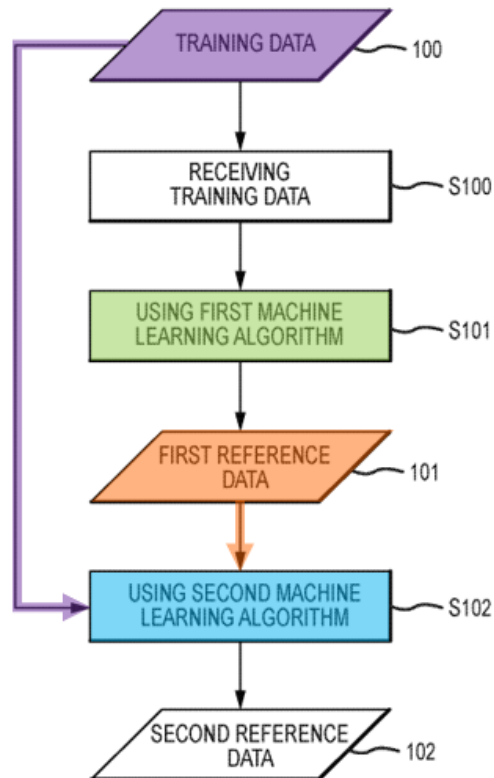


FIG.8

Ex. 1001, Fig. 8; *see also id.*, Fig. 9. “The second machine learning algorithm MLA2 thus gets the editing instructions to match or to approximate, and does not have to infer these from the matched input images as MLA1.” Ex. 1001, 17:14-17. I note that unless otherwise stated, I added the coloring to the figures in this declaration. Bolded text has also been added throughout my declaration for emphasis unless otherwise indicated.

VII. OVERVIEW OF THE PRIOR ART

A. Chalom

68. I understand that the Examiner was not aware of Chalom (Ex. 1005) during prosecution. Just like the '985 Patent, Chalom discloses using machine learning techniques for gaze correction. Ex. 1005, Title (“Eye contact correction in real time using machine learning”). As shown in Fig. 3 below, feature sets 312 (red) are generated based on the input image 212 (purple), both of which are provided to random forest classifier 306 (yellow) for determining the “optimal[] motion vector field 313,” which is used to warp the eye regions of the input image 212 (purple) to generate an “eye contact corrected image (ECCI) 215 [brown].” Ex. 1005, 9:13, 6:53-54.

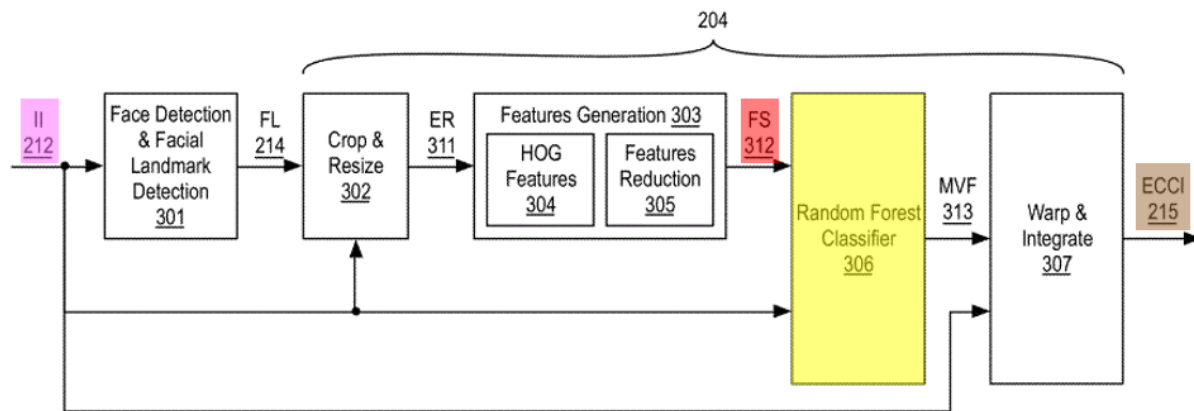


FIG. 3

Ex. 1005, Fig. 3.

69. The training of Chalom's random forest classifier 306 (yellow, above) is illustrated in Fig. 8 below.

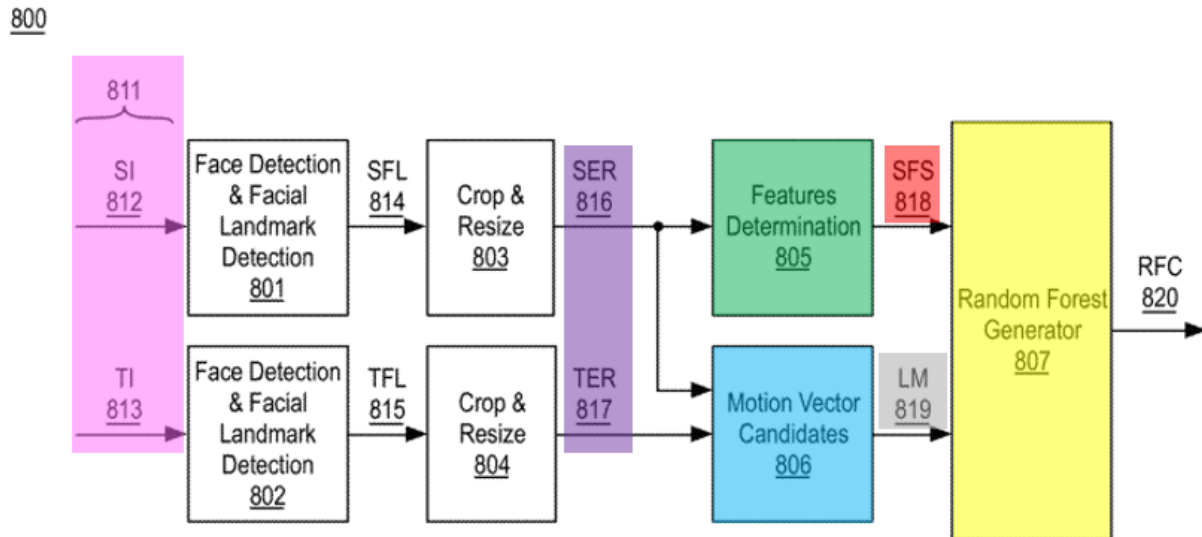


FIG. 8

Ex. 1005, Fig. 8. As shown, training images 811 (pink) “include pairs of source images 812 and target images 813.” Ex. 1005, 11:25-26. The source eye regions 816 are used by the features determination module 805 (green) to “determine source feature sets (SFS) 818 [red].” Ex. 1005, 13:8-11. Both source eye regions 816 and target eye regions 817 (purple) are used by the motion vector candidates module 806 (blue) to determine the candidate motion vectors and corresponding likelihood maps. Ex. 1005, 12:23-27. When training the random forest classifier, “each leaf of the tree of the random forest classifier may represent a likelihood map (e.g., a SAD map) for every pixel in the eye region and, at each branch of the tree, the training process may minimize the entropy between the likelihood maps

that have arrived at the branch.” Ex. 1005, 4:62-66.

B. Ganin

70. I understand that the Examiner was aware of Ganin (Ex. 1006) during prosecution. However, for reasons I will explain in Section XII.A, *infra*, Ganin is material to the patentability of at least some of the challenged claims and the US Examiner for the '985 Patent erred in a manner material to the patentability of these challenged claims by overlooking Ganin.

71. Ganin was published at least as early as September 17, 2016. This is indicated by Exhibit 1008, which is a printout of a SpringerLink webpage at: https://link.springer.com/chapter/10.1007/978-3-319-46475-6_20. As shown in Exhibit 1008, Ganin is a “Conference paper” from “Computer Vision – ECCV 2016” and was “First [published] Online: 17 September 2016.” I am familiar with ECCV, which refers to “European Conference on Computer Vision.” *See* Ex. 1050. According to Exhibit 1050, ECCV 2016 conference was held on October 11-14, 2016, in Amsterdam, The Netherlands. Exhibit 1008 shows that Ganin was one of the papers submitted to ECCV 2016. Ganin as provided in Exhibit 1006 is directly downloadable from the webpage as shown in Exhibit 1008.

72. Exhibit 1006 also provides Ganin’s copyright information, shown at the bottom of Exhibit 1006: “© Springer International Publishing AG 2016.”

Based on my experience, this indicates that Ganin was published by Springer International Publishing in 2016.

73. Ganin's public availability is also confirmed by the Wayback machine provided by Internet Archive, which I am familiar with. For example, as shown in Exhibit 1048 and its Exhibit A, authors of Ganin set up a project page for Ganin at least as early as September 5, 2016. As shown in Exhibit 1048 and its Exhibit B, Ganin was publicly accessible and downloaded from the author's project page at least as early as March 29, 2017.

74. Based on my experience, all these information indicates to me that Ganin was publicly available before the earliest priority date of the '985 Patent, and that Exhibit 1006 is a true and correct copy of Ganin and creates no suspicion about its authenticity.

75. As shown in Fig. 2 below, Ganin concatenates the input eye region, anchor points extracted from the eye region, and the desired angle into maps, which are then provided to the 2-scale flow NN that predicts a "warping field" (red) to correct the gaze of an input image. Ex. 1006, p. 4.

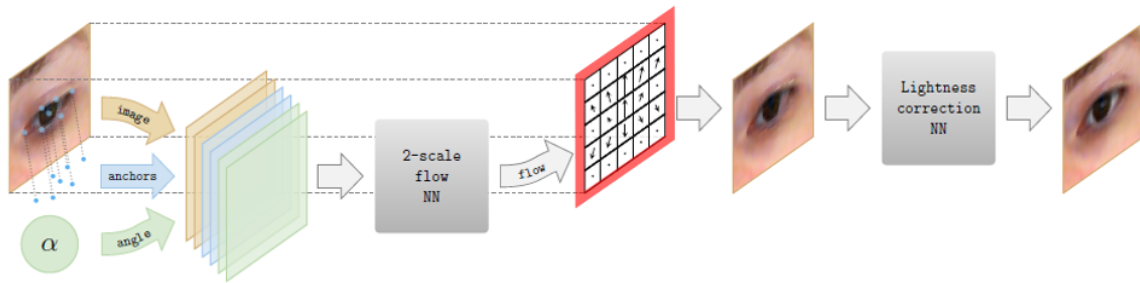


Fig. 2. The **proposed system** takes an input eye region, feature points (**anchors**) as well as a correction **angle** α and sends them to the multi-scale neural network (see Section 3.2) predicting a **flow** field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

Ex. 1006, Fig. 2 (colors original except for the red frame indicating the flow field). Ganin’s “multi-scale neural network” includes a coarse warping module and a fine warping module, both of which include a neural network. *Id.*, pp. 5-6. The coarse warping module generates the coarse warping field and the fine warping module generates the residual warping field. *Id.* The coarse warping field and the residual warping field are then combined together to warp the input eye region. *Id.*; Figs. 10(a), 10(c).

C. Aslan

76. I understand that the Examiner was not aware of Aslan (Ex. 1007) during prosecution. Aslan teaches a “process for jointly training multiple machine learning models.” Ex. 1007, Abstract. Figure 1 below illustrates Aslan’s teacher-student training architecture: the first machine learning algorithm 100 (green) is a

teacher model and the second machine learning algorithm 102 (blue) is a student model.

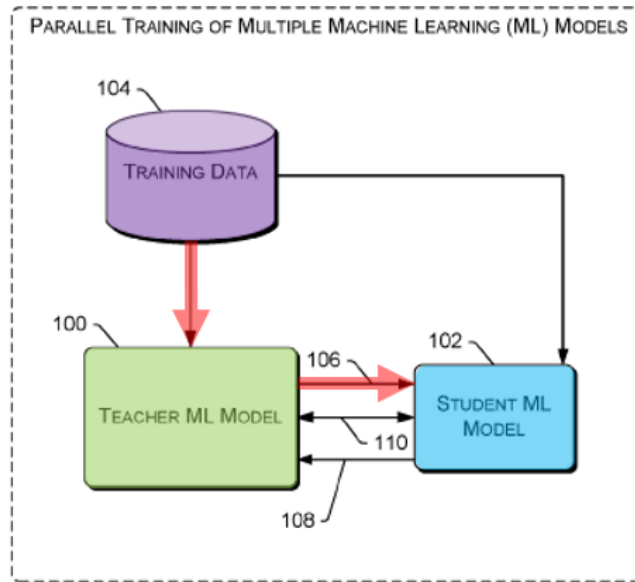


Fig. 1

Ex. 1007, Fig. 1. The training data 104 (purple) is used to train both the teacher model and the student model. *Id.*, ¶23. “Joint training of the first model 100 and the second model 102 involves training the models 100 and 102 in parallel such that at least one of the models 100 and/or 102 influences the training of the other model.” *Id.*, ¶26. This is implemented “by the path 106 [red] that goes from the training data 104 to the first model 100, and from the first model 100 to the second model 102,” such that “the training of the second model 102 can be influenced by what the first model 100 is learning from the training data 104.” *Id.*

D. Grau

77. I understand that the Examiner was not aware of Grau (Ex. 1009) during prosecution. As I will explain below, Grau discloses the well-known techniques of using machine learning to generate other image editing instructions, such as a filter field, a brightness adjustment field, or a texture blending field, as required by dependent claim 4 of the '985 Patent.

E. Pace

78. I understand that the Examiner was not aware of Pace (Ex. 1010) during prosecution. As I will explain below, Pace discloses the well-known use of Discrete Wavelet Transformation (DWT) for data compression, as required by dependent claim 7 of the '985 Patent.

F. Liu

79. I understand that the Examiner was not aware of Liu (Ex. 1016) during prosecution. As I will explain below, Liu discloses the well-known use of Principal Component Analysis (PCA) for data compression, as required by dependent claims 16 and 19 of the '985 Patent.

80. Liu was published at least as early as March 8, 2010. This is indicated by Exhibit 1051, which is a printout of a SpringerOpen webpage at: <https://asp-eurasipjournals.springeropen.com/articles/10.1155/2010/680623>. As shown in Exhibit 1051, Liu was “Published: 08 March 2010.” As shown in Exhibit 1051,

Liu was published in EURASIP Journal on Advances in Signal Processing, specifically in 2010. Liu as provided in Exhibit 1016 is directly downloadable from the webpage as shown in Exhibit 1051.

81. Exhibit 1016 also provides similar information confirming Liu's public availability date. For example, Exhibit 1016 shows the following on its first page:

Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2010, Article ID 680623, 6 pages
doi:10.1155/2010/680623

This confirms that Liu as shown in Exhibit 1016 was published in 2010 as part of Volume 2010 of EURASIP Journal on Advances in Signal Processing.

82. As shown in Exhibit 1048 and its Exhibit C, Liu was received on "6 December 2009," accepted on "16 January 2010," and published on "8 March 2010" by "EURASIP Journal on Advances in Signal Processing." Ex. C to Ex. 1048. Exhibit 1048 and its Exhibit C also include copyright information shown as "© Kui Liu et al. 2010," indicating that Liu was published in 2010.

83. As shown in Exhibit 1052, Liu is also publicly accessible at: <https://dl.acm.org/doi/abs/10.1155/2010/680623#>. According to Exhibit 1052, Liu was "Published: 01 February 2010."

84. Liu's public availability is also confirmed by the Wayback machine provided by Internet Archive, which I am familiar with. For example, as indicated in Exhibit 1048 and its Exhibit C, the link leading to Exhibit 1051 and the full text of Liu contained thereon were publicly accessible at least as early as May 14, 2017. *See* Ex. C to Ex. 1048.

85. Based on my experience, all these information indicates to me that Liu was publicly available before the earliest priority date of the '985 Patent, and that Exhibit 1016 is a true and correct copy of Liu and creates no suspicion about its authenticity.

VIII. CLAIM CONSTRUCTION

86. I understand that claim construction is the process of determining the meaning of a term or phrase in a patent claim. I understand that the proper construction of a term is how a PHOSITA would have understood the term based on its use in the claims, specification, and file history of the patent. I further understand that the claims are construed before the Board according to the same claim construction standard that applies in district courts. I have followed these principles in my analysis throughout this declaration. For the purpose of this declaration, I do not believe that any express constructions are needed and have applied the plain and ordinary meaning of all terms.

IX. ANTICIPATION

87. I have been instructed as to the definition of “anticipation” in the context of the patent laws.

88. I understand that anticipation of a claim requires that every element of a claim be disclosed expressly or inherently in a single prior art reference, in combination, as claimed. I understand that a single prior art reference may anticipate claims without expressly disclosing a feature of the claimed invention if that feature is necessarily present, or inherent, in that reference. I understand that a reference is read from the perspective of one of ordinary skill in the art at the time of the invention.

X. OBVIOUSNESS

89. I have been instructed as to the definition of “obviousness” in the context of the patent laws.

90. It is my understanding that obviousness is a question of law based on underlying factual issues including the content of the prior art and the level of skill in the art. I understand that for a single reference or a combination of references to render the claimed invention obvious, a person of ordinary skill in the art must have been able to arrive at the claims by altering or combining the applied references.

91. I also understand that when considering the obviousness of a patent claim, one should consider whether a teaching, suggestion, or motivation to combine the references exists to avoid impermissibly applying hindsight when considering the prior art. I understand this test should not be applied rigidly, but that the test can be important to avoid such hindsight.

XI. OPINIONS REGARDING PATENTABILITY

92. In my opinion, claims 1-3, 5, 10, 12-14, and 18 are anticipated by Ganin. Additionally, claims 16 and 19 are rendered obvious by Ganin further in view of Liu.

93. Moreover, in my opinion, claims 1-3, 5-6, 8-14, and 16-19 are rendered obvious by the Chalom-Aslan combination. Additionally, claim 4 is rendered obvious by the Chalom-Aslan combination further in view of Grau. Claim 7 is rendered obvious by the Chalom-Aslan combination further in view of Pace.

94. I have based my opinion on the following references:

- **Ganin:** “Deepwarp: Photorealistic image resynthesis for gaze manipulation” authored by Yaroslav Ganin *et al.* (“Ganin,” Ex. 1006) was published at least as early as **September 17, 2016**. *See* Ex. 1008; *supra* Section VII.B. I understand that Ganin is prior art to the ’985 Patent.

- **Chalom:** U.S. Patent No. 10,664,949 titled “Eye contact correction in real time using machine learning” to Edmond Chalom and Or Shimshi (“Chalom,” Ex. 1005), was filed **April 22, 2016**, published October 26, 2017, granted May 26, 2020. I understand that Chalom is prior art to the ’985 Patent.

- **Aslan:** U.S. Patent Publication No. 2017/0132528 A1 titled “Joint model training” to Ozlem Aslan *et al.* (“Aslan,” Ex. 1007) was filed June 28, 2016 and published **May 11, 2017**. I understand that Aslan is prior art to the ’985 Patent.

- **Grau:** U.S. Patent Publication No. 2018/0158246 A1 to Grau *et al.* (“Grau,” Ex. 1009) is based on an application filed **December 7, 2016**. I understand that Grau is prior art to the ’985 Patent.

- **Pace:** U.S. Patent No. 8,942,283 to Charles P. Pace (“Pace,” Ex. 1010) issued on **January 27, 2015**. I understand that Pace is prior art to the ’985 Patent.

- **Liu:** Liu is an article titled “Optical Flow and Principal Component Analysis-Based Motion Detection in Outdoor Videos” (“Liu,” Ex. 1016) that was published at least as early as **March 8, 2010**. Ex. 1048; *supra* Section VII.F. I understand that Liu is prior art to the ’985 Patent.

95. In my opinion, the challenged claims of the '985 Patent are unpatentable based on the following grounds:

- (1) Claims 1-3, 5, 10, 12-14, and 18 are unpatentable under § 102 as anticipated by Ganin;
- (2) Claims 16 and 19 are unpatentable under § 103 as obvious over Ganin in view of Liu;
- (3) Claims 1-3, 5-6, 8-14, and 16-19 are unpatentable under § 103 as obvious over Chalom in view of Aslan;
- (4) Claim 4 is unpatentable under § 103 as obvious over Chalom in view of Aslan and Grau; and
- (5) Claim 7 is unpatentable under § 103 as obvious over Chalom in view of Aslan and Pace.

XII. MY OPINIONS AS TO THE CHALLENGED CLAIMS OF THE '985 PATENT

A. My Opinions as to Claims 1-3, 5, 10, 12-14, and 18 in view of Ganin alone

1. Claim 1

96. In my opinion, claim 1 is unpatentable because it is anticipated by Ganin. Claim 1 of the '985 Patent recites the following. I address each of these limitations in my analysis below.

1[P] A method of generating reference data for adjusting a digital representation of a head region, the method comprising:

1[A] receiving training data comprising:

1[B] a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

1[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

1[D] using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and

1[E] using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

1[F] wherein the first reference data comprise first editing instructions for a range of possible configurations of the target feature and first selection instructions for selecting editing instructions for a particular input patch from the

first editing instructions based on the configuration of the target feature of the input patch; and

1[G] wherein the second reference data comprise second editing instructions for a range of possible configurations of the target feature and second selection instructions for selecting editing instructions for a particular input patch from the second image editing instructions based on the configuration of the target feature of the input patch.

a. 1[P]

97. If the preamble is limiting, it is my opinion that Ganin discloses it.

98. In my opinion, Ganin discloses a training process that generates “reference data for adjusting a digital representation of a head region,” e.g., for gaze correction.

99. The “core of [Ganin’s] approach is the prediction of the **warping field** using a deep convolutional network.” Ex. 1006, p. 13. Ganin’s proposed system is shown in Figure 2 reproduced below.

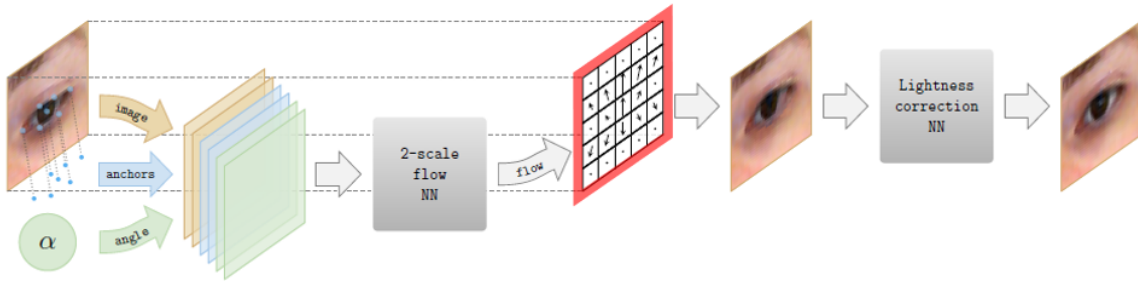


Fig. 2. The **proposed system** takes an input eye region, feature points (**anchors**) as well as a correction **angle** α and sends them to the multi-scale neural network (see Section 3.2) predicting a **flow field**. The flow field is then applied to the input image to produce an image of a **redirected eye**. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

Ex. 1006, Fig. 2 (colors original except for the red frame). As shown, given an input eye region (“image,” brownish) and other required information, the proposed “multi-scale neural network” (“2-scale flow NN,” gray) predicts a “warping field” (red)/“flow field”/“pixel-flow field”/“two-channel map,” which “is then **applied to the input image** to produce an image of a **redirected eye**.” *Id.*, p. 4. I note that Ganin uses “warping field,” “flow field,” “pixel-flow field,” “two-channel map” interchangeably, and thus I also use these terms interchangeably herein as to Ganin.

100. Ganin also discloses that “[t]he architecture is **trained** end-to-end in a supervised way using a specially collected dataset that depicts the change of the appearance under gaze redirection in real life.” Ex. 1006, p.2. Specifically, Ganin’s training process takes “**training pairs**... [of] two images with different gaze directions,” and passes them through the “multi-scale neural network” such

that the weights of the nodes within the “multi-scale neural network” are optimized by minimizing the “ ℓ_2 -distance between the synthesized output O_{output} [e.g., gaze corrected eye regions, such as eye regions of the first images of the training pairs **warped by the predicted warping fields**] and the ground-truth O_{gt} [e.g., target eye regions, such as eye regions of the second images of the training pairs].” *Id.*, p.8.

101. In view of (i) Ganin’s training process comparing warped eye regions with target eye regions and (ii) the trained “multi-scale neural network” being able to generate a warping field given an eye region input, it is my opinion that Ganin’s warping fields are expressly taught or, at a minimum, necessarily generated during the training process, which thus discloses the claimed “reference data.” *Infra* Sections XII.A.1.e and XII.A.1.f.

102. Therefore, in my opinion, Ganin discloses “[a] method of generating reference data [e.g., warping fields generated during training; “first reference data” in Limitation 1[D]; “second reference data” in Limitation 1[E]] for adjusting a digital representation of a head region [e.g., gaze correction].”

b. 1[A]

103. As shown in Figure 2 below, Ganin discloses that its proposed multi-scale neural network receives three types of input data: (1) an input eye region

(“image,” brownish), (2) feature points (“anchors,” blue), and (3) a correction angle (“angle,” green).

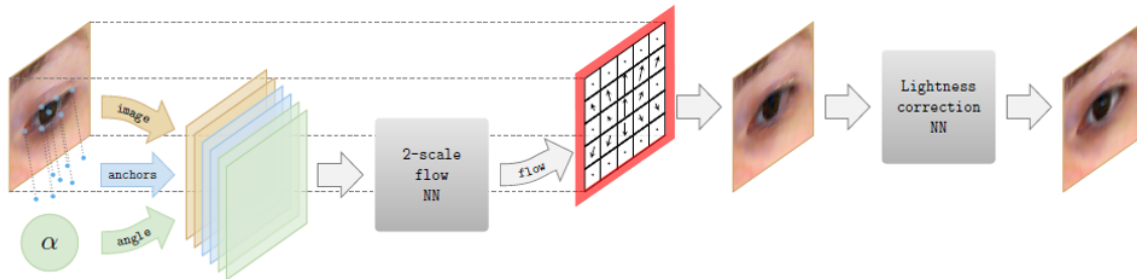


Fig. 2. The **proposed system** takes an input eye region, feature points (**anchors**) as well as a correction **angle** α and sends them to the multi-scale neural network (see Section 3.2) predicting a **flow** field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

Ex. 1006, Fig. 2 (colors original except for the red frame), Fig. 10(a). Since Ganin’s neural network (“NN”) is “trained end-to-end in a supervised way,” the training of the NN also includes these three types of inputs, plus the ground-truth eye regions: “[t]he model is **trained on pairs of images** corresponding to eye appearance **before and after the redirection.**” *Id.*, pp. 2 and 4, p. 5 (referencing “**pairs of images**”). Ganin discloses “extract[ing] the image parts around each of the eye and resiz[ing] them to characteristic scale” for both the “input image” and the “output [or target] image,” and “[t]he resulting image pair serves as a **training example for the learning procedure** (Figure 4-Right).” *Id.*, p. 5.

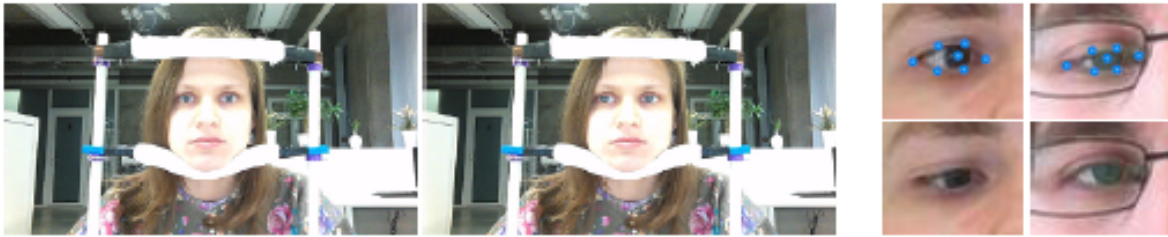


Fig. 4. Left – dataset collection process. Right – examples of two training pairs (input image with superimposed feature points on top, output image in the bottom).

Id., Fig. 4.

104. Therefore, in my opinion, Ganin discloses “receiving training data [e.g., eye regions extracted from image pairs].”

c. 1[B]

105. Ganin discloses that “[t]he model is **trained on pairs of images** corresponding to eye appearance **before** and after **the redirection.**” Ex. 1006, p. 4. Ganin further discloses extracting the eye region by aligning the eyes, “comput[ing] a tight axis-aligned bounding box B' of the points in the **input image,**” and “cut[ting] out the interior of the estimated box from the **input image,** and also from the **output image** of the pair (using exactly the same bounding box coordinates).” *Id.*, p. 5. Thus, “[t]he resulting image pair serves as **a training example for the learning procedure** (Figure 4-Right).” *Id.* Thus, in my opinion, Ganin discloses that all training images, including the input images and the output images, would have contained eye regions and the eye region extraction process is performed for each image pair.

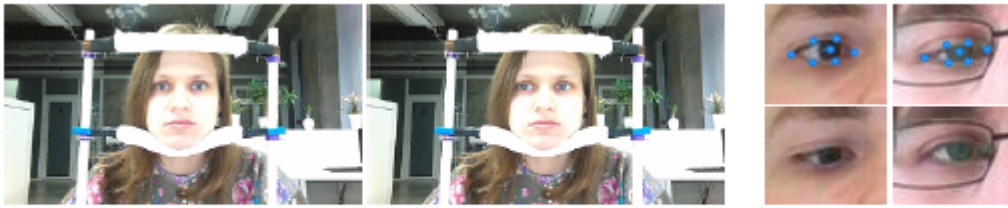


Fig. 4. Left – dataset collection process. Right – examples of two training pairs (input image with superimposed feature points on top, output image in the bottom).

Id., Fig. 4.

106. Therefore, in my opinion, Ganin discloses “a set of input patches [e.g., extracted eye regions of input images corresponding to eye appearance before gaze redirection], each input patch comprising a target feature [e.g., eye region] of a digital representation [e.g., image] of a head region prior to adjustment [e.g., before gaze redirection] of the digital representation of the head region, wherein the target feature is the same for each input patch.”

d. 1[C]

107. Ganin discloses that “[t]he model is **trained on pairs of images** corresponding to eye appearance before and **after the redirection.**” Ex. 1006, p. 4. “The head pose, the lighting, and all other nuisance parameters are (approximately) the **same between the two images in the pair.**” *Id.*, p. 5. In my opinion, a PHOSITA would have understood that “pair” means “two” and that both images in each pair of images include “eye appearance” (or eye regions), with **one showing the eye region before gaze redirection and the other showing the same**

eye region after gaze redirection. *See also, e.g.,* Ex. 1006, p.5 (“At training time, our dataset allows us to mine **pairs of images** containing eyes of the same person **looking in two different directions separated by a known angle α .**”). The one-to-one correspondence is further illustrated in Fig. 4 below (“Right – examples of two training pairs” with “input image [] on top, [and corresponding] output image in the bottom”).

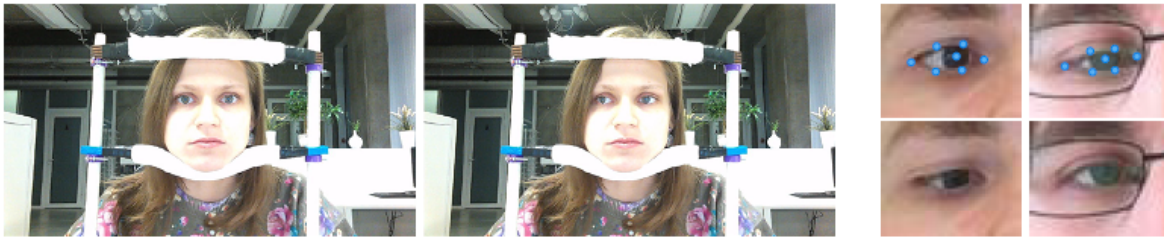


Fig. 4. Left – dataset collection process. Right – examples of two training pairs (input image with superimposed feature points on top, output image in the bottom).

Ex. 1006, Fig. 4. As shown, for each input image, there is a corresponding output image having a known difference in gaze direction.

108. Therefore, it is my opinion that Ganin’s output images (and eye regions thereof) are in one-to-one correspondence with the input images (and the eye regions thereof).

109. Ganin further discloses extracting the eye region by aligning the eyes, “comput[ing] a tight axis-aligned bounding box B' of the points in the **input image**,” and “cut[ting] out the interior of the estimated box from the **input image**,

and also from the **output image** of the pair (**using exactly the same bounding box coordinates**)." Ex. 1006, p. 5. Thus, "[t]he resulting image pair serves as a **training example for the learning procedure** (Figure 4-Right)." *Id.* In my opinion, this further confirms that Ganin's output images are in one-to-one correspondence with its input images because in order to "cut out the interior of the estimated box [i.e., eye region] from the output image of the pair [] using exactly the same bounding box coordinates" (*Id.*), the output image necessarily has an eye region located at approximately the same location within the output image as the corresponding eye region in the input image.

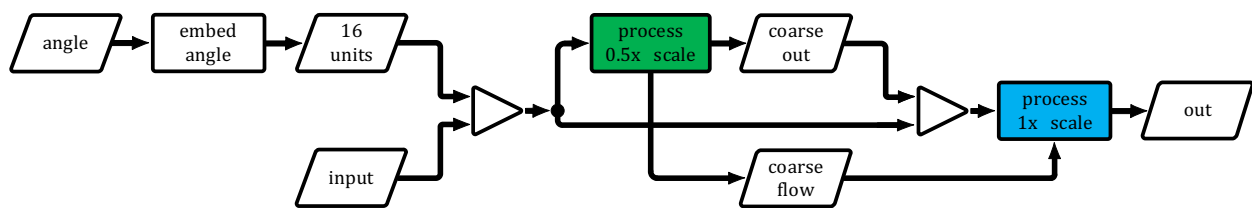
110. Therefore, in my opinion, Ganin discloses "a set of output patches [e.g., extracted eye regions of input images corresponding to eye appearance after gaze redirection] in one-to-one correspondence with the input patches, each output patch comprising the target feature [e.g., eye region] of the digital representation of the head region after adjustment [e.g., after gaze redirection] of the digital representation of the head region."

e. 1[D]

111. Ganin proposes "warping the input image." Ex. 1006, p. 4. Before moving on, I would like to note that Ganin discusses "input images" in two different contexts. During training, an input image refers to an image with gaze direction to be corrected, which is part of the training data. *See* Ex. 1006, Fig. 4

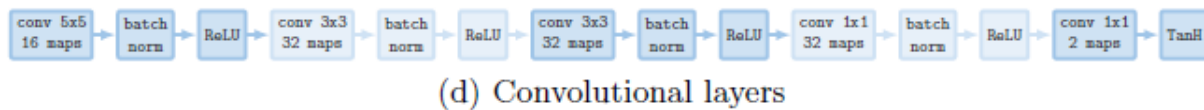
(“Right – examples of two training pairs (input image with superimposed feature points on top, output image in the bottom)”). During training, the “output images” in Ganin are also provided as ground truth to train the machine learning algorithm (MLA). This is also consistent with how the ’985 Patent itself uses “input patches” and “output patches.” During runtime, an input image also refers to an image with gaze direction to be corrected, which, however, is not part of the training data, and the corresponding “output image” is not provided—rather, the corresponding “output image” is what the trained MLA is expected to provide. With these contexts in mind, I now discuss Ganin’s disclosures with respect to 1[D].

112. Ganin’s warping field to warp the image “is predicted in two stages in a coarse-to-fine manner,” and the “coarse warping” is implemented by a $0.5\times$ -scale processing module and the “fine warping” is implemented by a $1\times$ -scale processing module. Each of these “warping modules” warps the input image by predicting a pixel-flow field and applying the predicted pixel-flow field to the input image. Ex. 1006, p. 4, Fig. 10 caption. The “high-level pipeline” is shown in Fig. 10(a), reproduced below.



(a) High-level pipeline

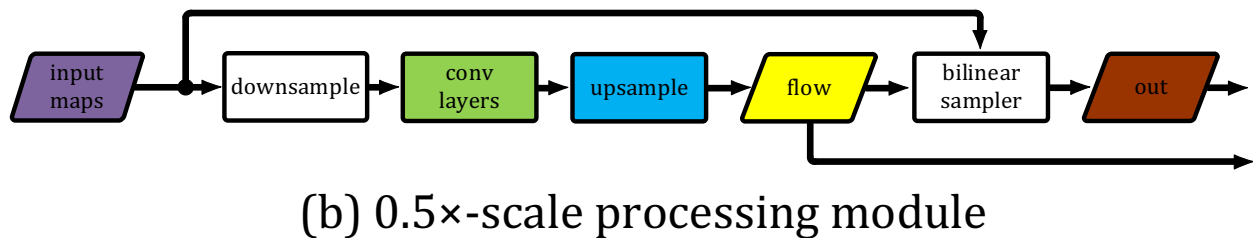
Id., Fig. 10(a) (0.5 $\times$ -scale processing module in green; 1 $\times$ -scale processing module in blue). I note that I have added coloring for Ganin's Figures 10(a), (b), and (c) unless otherwise noted. According to Ganin, each of the two processing modules is a MLA, comprising a NN having "convolutional layer[s]." *See, e.g.*, Ex. 1006, p. 4 ("multi-scale neural network"); p. 17 ("**The architectures of [the convolutional neural network within] the two warping modules are same (modulo the number of input maps) and are shown in Figure 10(d).**"); p. 5 ("**Coarse warping. The last convolutional layer of the first (half-scale) warping module produces a pixel-flow field (a two-channel map)**"); p. 6 ("**Fine warping. In the fine warping module, the rough image estimate O_{coarse} and the upsampled low-resolution flow $D_{\text{coarse}}(I, \alpha)$ are concatenated with the input data... and sent to the 1 $\times$ -scale network**").



Id., Fig 10(d) (illustrating the convolutional layers for both warping modules). I note here that the convolutional layers in Ganin's Figure 10(d) show the architecture of the NN, which is one component of Ganin's two warping modules. The NN is used within each module to form a prediction of the pixel-flow field. In the coarse 0.5 $\times$ -scale warping module, the NN is used to form an initial prediction of the pixel-flow field (i.e., downsampled low-resolution pixel-flow field in coarse

0.5×-scale warping module) that is upsampled to get the final coarse pixel-flow field that is predicted by the coarse warping module and applied to warp the input image producing a “coarse” warped output image. In the fine 1×-scale warping module, the NN is used to predict a residual pixel-flow field that is added to the coarse pixel-flow field that is generated by the 0.5×-scale coarse warping module to get the final pixel-flow field. *Id.*, pp. 5-6.

113. Ganin’s coarse warping module, or 0.5×-scale processing module, is further detailed in Fig. 10(b), reproduced below.



Ex. 1006, Fig. 10(b).

114. As shown, Ganin’s coarse warping module generates two outputs. One is the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ (yellow). “The last convolutional layer [green] of the first (half-scale) warping module produces a pixel-flow field (a two-channel map), which is then up-sampled [blue] [to generate the coarse pixel-flow field] $\mathbf{D}_{\text{coarse}}(I, \alpha)$ and applied to warp the input image.” Ex. 1006, p. 5. The pixel-flow field instructs how pixels of the input image is warped and in my opinion, is therefore an “editing instruction[] for adjusting the digital representation of the

head region” as required by 1[D]. Ex. 1001, 8:47-50 (“For an input and output image of each individual, there is derived a displacement vector field that represents the transformation of the image patch in the input image required to obtain the image patch of the output image”), 9:11-12 (“displacement vector field F is referred to as “flow field””), 9:19-29 (“editing instructions... which can be used additionally or as an alternative to the displacement vector field in any arrangements disclosed herein... could be the output of a convolution of the patch...”). The coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ instructs how the pixels of the input image may be warped to generate the other output of the coarse warping module “coarse estimate O_{coarse} ” (brown):

$$O_{\text{coarse}} = \mathbf{S}(I, \mathbf{D}_{\text{coarse}}(I, \alpha)) . \quad (1)$$

Ex. 1006, p. 5, Eq. 1. Therefore, in my opinion, both the “pixel-flow field” and the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ up-sampled therefrom are “editing instructions for adjusting the digital representation of the head region” required by 1[D].

115. During both training and runtime, Ganin’s coarse warping module receives “input maps,” which include “input images” before gaze redirection, “the desired redirection angle,” and “feature points” generated “from the raw input pixels.” Ex. 1006, p. 6. During training, Ganin teaches using “a regular ℓ_2 -distance between the synthesized output O_{output} and the ground-truth O_{gt} as the

objective function.” *Id.*, p. 8. In my opinion, a PHOSITA would have understood that the synthesized output O_{output} could refer to both coarse estimate O_{coarse} (output of the coarse warping module) and final output O (output of the fine warping module). Accordingly, the ground-truth O_{gt} refers to the “output images” having redirected gazes, which are provided to the both the coarse warping module (to calculate the ℓ_2 -distance between coarse estimate O_{coarse} and O_{gt}) and the fine warping module (to calculate the ℓ_2 -distance between final output O and O_{gt}) during training.

116. To the extent Patent Owner argues that Ganin discloses “end-to-end” training (Ex. 1006, p.2), that does not affect my analysis. End-to-end training refers to training the whole system together using one cost function, as opposed to, for example, training the coarse warping module using one cost function and **then** separately training the fine warping module using the same or a different cost function. The fact that Ganin’s two warping modules are trained “end-to-end” does not necessarily mean that the ground-truth O_{gt} is provided only to the last warping module, i.e., fine warping module. A PHOSITA would have understood that, even in end-to-end training, the ground-truth O_{gt} can be provided to both coarse warping module and fine warping module. In doing so, the ℓ_2 -distance is calculated for both O_{coarse} vs O_{gt} (e.g., coarse distance) and final output O vs O_{gt} (e.g., final distance). Then, the cost function may be established by combining the

coarse distance and the final distance, for example, by doing a weighted sum. This was well known to a PHOSITA. Aslan also illustrates a similar way of constructing a cost function for training teacher and student modules. Ex. 1007, ¶¶35-36, Eq. 2. In this way, Ganin's system is trained in an end-to-end manner with the ground-truth O_{gt} being provided for both the coarse warping module and the fine warping module.

117. Thus, it is my opinion that Ganin's coarse warping module is trained using the claimed "training data" that includes both "input images" and "output images."

118. In order to generate "coarse estimate O_{coarse} " during training, it is my opinion that the corresponding pixel-flow field and coarse warping field $\mathbf{D}_{coarse}(I, \alpha)$ up-sampled therefrom are expressly taught or, at a minimum, necessarily generated.

119. Moreover, as I will explain in Section XII.A.1.f, Ganin teaches that the fine warping module, or $1\times$ -scale processing module, receives as part of its input the coarse warping field $\mathbf{D}_{coarse}(I, \alpha)$. In order to enable the end-to-end training that is disclosed by Ganin, it is my opinion that Ganin's fine warping module receives during training such coarse warping fields generated from the training data.

120. Therefore, in my opinion, Ganin's coarse warping module generates "editing instructions for adjusting the digital representation of the head region" during training.

121. Finally, it is also my opinion that Ganin's editing instructions generated during training of the coarse warping module provide adjustments "for a range of possible digital representations of the head region."

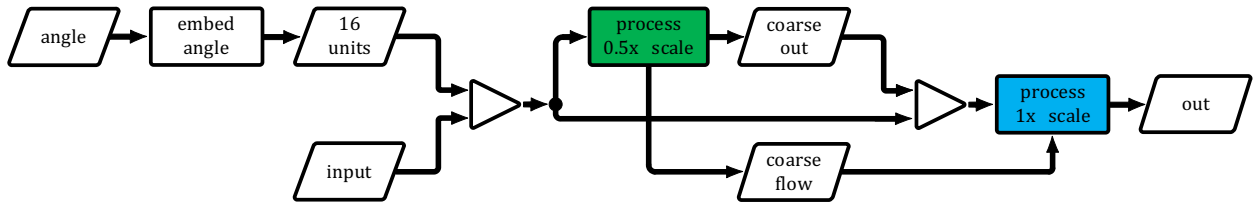
122. According to the '985 Patent, "a range of possible digital representations of the head region" could be "digital representations representing different states of the head, such as different positions and/or orientations." Ex. 1001, 14:47-54. In my opinion, Ganin discloses the same. For example, when preparing the training data, Ganin discloses that "[f]or each person we record 2 – 10 sequences, **changing the head pose** and light conditions **between different sequences**. Training pairs are collected, **taking two images** with different gaze directions **from one sequence**." Ex. 1006, p. 8. After "exclud[ing] bad shots," the ultimate training data includes a "dataset of 33 persons and 98 sequences." *Id.* Since there are more sequences than persons, it is my opinion that the training data expressly (or at a minimum, necessarily) includes multiple image pairs of the same person taken from different sequences, i.e., having different head poses. *Id.* ("we collect our own dataset (Figure 4)... we record [for each person] several images with eyes looking in **different fixed directions** (about 200 for one video sequence)

[]. For each person we record 2-10 sequences, **changing the head pose and light conditions** between different sequences. Training pairs are collected, taking two images with **different gaze directions** from one sequence.”); p. 9 (“**Sampling training pairs**. We found that biasing the selection process for **more difficult and unusual head poses** and bigger redirection angles improved the results.”); p. 13 (“Our system is reasonably robust against different head poses (e.g., see Figure 3)”); Figure 1; Figure 1 caption (“Gaze redirection with our model trained for vertical gaze redirection ... [with] the desired redirection angle [] varying between -15 and +15 degrees []).”).

123. Therefore, in my opinion, Ganin discloses “using a first machine learning algorithm [e.g., coarse warping module including the $0.5\times$ -scale neural network] to generate first reference data [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$] using the training data [e.g., “input maps” including input images, and output images as ground truth], the first reference data comprising editing instructions [e.g., pixel-flow fields and coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ up-sampled therefrom] for adjusting the digital representation of the head region [e.g., gaze redirection] for a range of possible digital representations of the head region [e.g., different head poses, and/or different light conditions, and/or different gaze directions].”

f. 1[E]

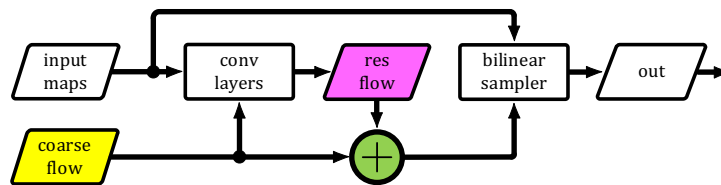
124. As I mentioned above, Ganin proposes generating warping fields (i.e., editing instructions) in a “coarse-to-fine manner,” which includes a $0.5\times$ -scale coarse warping module (MLA1) and a $1\times$ -scale fine warping module (MLA2), both of which are based on NNs. Ex. 1006, pp. 4, 5, 6.



(a) High-level pipeline

Id., Fig. 10(a) ($0.5\times$ -scale processing module in green; $1\times$ -scale processing in blue).

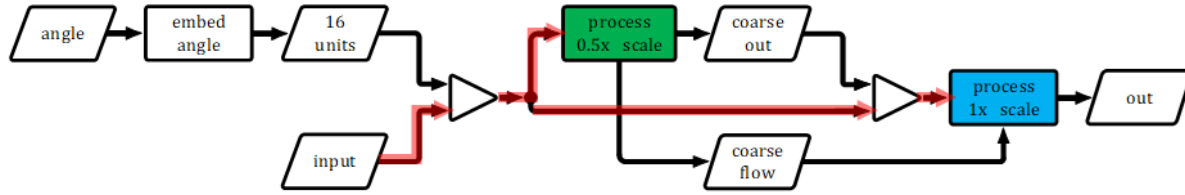
125. Ganin’s fine warping module is further detailed in Fig. 10(c), reproduced below.



(c) $1\times$ -scale processing module

Id., Fig. 10(c). As shown, Ganin’s fine warping module takes two inputs during runtime. First is the input maps. *See, e.g., Id.*, p. 5 (“Each of the two warping modules takes as an input the image, the position of the feature points, and the redirection angle.”). This is also confirmed by the “high-level pipeline” illustrating

both NN-based warping modules:



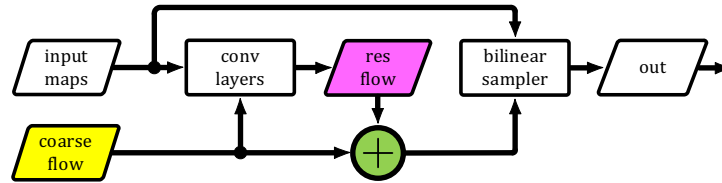
(a) High-level pipeline

Id., Fig. 10(a) (showing “input” being received (red) by both 0.5×-scale processing module (green) and 1×-scale processing module (blue)). The other input to Ganin’s 1×-scale NN is the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated by the coarse warping module. It is my opinion that these two inputs are provided to Ganin’s fine warping module during both runtime and training.

126. In addition, Ganin teaches using “a regular ℓ_2 -distance between the synthesized output O_{output} and the ground-truth O_{gt} as the objective function.” Ex. 1006, p. 8. The ground-truth O_{gt} refers to the “output images” having redirected gazes, which are provided to the fine warping module during training. Thus, it is my opinion that Ganin’s fine warping module is trained using the claimed “training data” that includes both “input images” and “output images.”

127. With respect to Fig. 10(c), Ganin further discloses that the fine (1×-scale) warping module’s NN predicts “residual flow” $\mathbf{D}_{\text{res}}$ (pink), “another two-

channel flow” that is added (green) to the coarse warping field $\mathbf{D}_{\text{coarse}}$ (yellow) to obtain the final predicted warping field $\mathbf{D}(I, \alpha)$.



(c) $1\times$ -scale processing module

Ex. 1006, Fig. 10(c). This addition is reflected here:

$$\mathbf{D}(I, \alpha) = \mathbf{D}_{\text{coarse}}(I, \alpha) + \mathbf{D}_{\text{res}}(I, \alpha, O_{\text{coarse}}, \mathbf{D}_{\text{coarse}}(I, \alpha)), \quad (3)$$

Id., Eq. 3. Such final warping field $\mathbf{D}(I, \alpha)$ is used to warp the original input image “to obtain the final output”:

$$O = S(I, \mathbf{D}(I, \alpha)) . \quad (4)$$

Id., p. 6, Eq. 4. Thus, it is my opinion that both the “residual flow” $\mathbf{D}_{\text{res}}$ and the final warping field $\mathbf{D}(I, \alpha)$ are “for adjusting the digital representation of the head region.” It is also my opinion that the “residual flow” $\mathbf{D}_{\text{res}}$ and the final warping field $\mathbf{D}(I, \alpha)$ are generated by the fine warping module during both runtime and training.

128. I would like to note that while Ganin does not explicitly disclose applying the “residual flow” $\mathbf{D}_{\text{res}}$ directly to the input eye region, that is not required by 1[E] either. 1[E] requires “the second reference data comprising editing instructions **for adjusting** the digital representation of the head region for a

range of possible digital representations of the head region.” Thus, by optimizing, augmenting, or fine-tuning the coarse warping field $\mathbf{D}_{\text{coarse}}$, Ganin’s “residual flow” $\mathbf{D}_{\text{res}}$ is used “for adjusting” the gaze direction for the input eye region.

129. For the same reasons as I explained in Section XII.A.1.e, it is my opinion that Ganin’s editing instructions generated during training of the fine warping module also provide adjustments “for a range of possible digital representations of the head region” because both the coarse and fine warping modules use the same training data having different poses.

130. Therefore, in my opinion, Ganin discloses “using a second machine learning algorithm [e.g., fine warping module including the $1\times$ -scale NN] to generate second reference data [e.g., residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$] using the same training data as the first machine learning algorithm [e.g., “input maps” including input images, and output images as ground truth] and the first reference data generated by the first machine learning algorithm [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$], the second reference data comprising editing instructions [e.g., residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$] for adjusting the digital representation of the head region [e.g., gaze redirection] for a range of possible digital representations of the head region [e.g., different head poses].”

g. 1[F]

131. In my opinion, Ganin discloses that its pixel-flow fields and coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated during training are “for a range of possible configurations of the target feature.” When collecting training data, Ganin “record[s] several images with eyes looking in different fixed directions” and therefore resulting in “continuously varying [] angle.” Ex. 1006, p.8. Moreover, since Ganin’s trained system is able to perform gaze redirection for a “redirection angle... selected **arbitrarily from a certain range**” (Ex. 1006, Abstract), it is my opinion that the training data necessarily includes images with different gaze directions. Accordingly, the “first editing instructions,” e.g., pixel-flow fields and coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated during training using such training data would be “for a range of possible configurations of the target feature,” e.g., different gaze directions.

132. In my opinion, Ganin also discloses that “the first reference data comprise... first selection instructions for selecting editing instructions for a particular input patch from the first editing instructions based on the configuration of the target feature of the input patch.” As I will discuss in Section XII.C.2.f *infra*, to the extent that the recitations of intended use in this clause are limiting, it is my opinion that Ganin discloses them.

133. Specifically, the training of Ganin's coarse warping module generates not only coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ but also the trained 0.5×-scale NN, represented by the weights of neurons of the 0.5×-scale NN. During runtime, the trained 0.5×-scale NN receives an input patch having an eye region before gaze redirection and uses these weights of neurons to generate a corresponding coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ for redirecting the input gaze by a specified angle. Ex. 1006, p. 5.

134. In addition, it is my opinion that the pixel-flow fields and the coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated **during runtime** are **selected from** the first editing instructions, i.e., the pixel-flow fields and the coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated **during training** and which are typically represented by a warping-field generating function that is learned by the NN. This is because during training, the training set covers a wide range of eye regions and to-be-warped pixels, and the corresponding coarse warping fields generated for each of the wide range of eye regions cover all the possible coarse warping fields for various input eye regions. During training, the NN can learn the corresponding warping-field generating function that covers all possible warping fields that can be generated for the considered task. During runtime, given a particular input eye region, the machine learning based module (the coarse warping module

specifically) selects a coarse warping field from among all of the possible coarse warping fields that can be generated by the trained network.

135. According to the '985 Patent, “[t]he configuration of the target feature of each input patch may be represented by a feature vector derived from plural local descriptors (e.g. local image descriptors) of the input patch.” Ex. 1001, 15:20-26. In my opinion, Ganin discloses the same.

136. As shown in Figure 2 below, Ganin discloses generating feature maps and anchor maps for an input image, which are concatenated with the RGB channels of the input image to form an “input map stack” (brownish/blue/green, i.e., configuration of the target feature) for predicting the flow field for the input image.

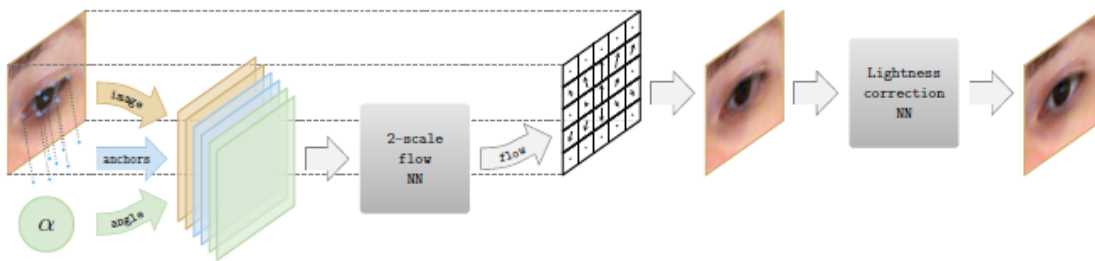


Fig. 2. The proposed system takes an input eye region, feature points (anchors) as well as a correction angle α and sends them to the multi-scale neural network (see Section 3.2) predicting a flow field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

Ex. 1006, Fig. 2 (colors original). In other words, Ganin’s configuration of the target feature of each input image (e.g., different pose orientations and/or gaze directions contained in each input image) is represented by the corresponding input

map stack.

137. Therefore, in my opinion, Ganin discloses that “the first reference data comprise first editing instructions [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$] for a range of possible configurations of the target feature [e.g., different gaze directions] and first selection instructions [e.g., trained 0.5 $\times$ -scale NN, represented by the weights of its neurons] for selecting editing instructions [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$] for a particular input patch from the first editing instructions [e.g., coarse warping fields $\mathbf{D}_{\text{coarse}}(I, \alpha)$ generated during training or representation of these through the warping-fields generating function that is learned by the network during the training phase] based on the configuration of the target feature [e.g., input map stack] of the input patch.”

h. 1[G]

138. Ganin discloses that “[t]he architectures of [the NNs within] the two warping modules are same (modulo the number of input maps) and are shown in Figure 10(d).” Ex. 1006, p. 17, Fig. 10(d). As I explained earlier, both receive the same training data and both generate warping fields, only at different scales. Thus, for reasons I explained in Section XII.A.1.g, it is my opinion that Ganin discloses that the residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ generated during training are also “for a range of possible configurations of the target feature.” Similar to the “first selection instructions” I discussed in Section XII.A.1.g, it is my opinion

that Ganin's "second reference data comprise... second selection instructions [e.g., trained $1\times$ -scale NN, represented by the weights of its neurons] for selecting editing instructions [e.g., residual flows $\mathbf{D}_{\text{res}}$ and final warping fields $\mathbf{D}(I, \alpha)$] for a particular input patch from the second image editing instructions [e.g., residual flows $\mathbf{D}_{\text{res}}$ and final warping fields $\mathbf{D}(I, \alpha)$ generated during training or representation of these through the warping-fields generating function that is learned by the network during the training phase] based on the configuration of the target feature [e.g., input map stack] of the input patch."

139. Therefore, in my opinion, Ganin discloses this limitation.

140. Because Ganin discloses each and every limitation of claim 1, it is my opinion that claim 1 is anticipated and therefore unpatentable.

2. Claim 2

141. In my opinion, claim 2 is unpatentable because it is anticipated by Ganin. Claim 2 of the '985 Patent recites the following. I address each of these in my analysis below.

2[P] The method of claim 1, wherein

2[A] the configuration of the target feature of each input patch is represented by a feature vector derived from plural local descriptors of the input patch, and;

2[B] the first and second selection instructions define how the feature vector is used to select editing instructions for the input patch.

142. First, it is my opinion that Ganin discloses “plural local descriptors of the input patch.”

143. According to the '985 Patent, “local descriptors []represent[] **information about a local region** in a patch[], which... may be referred to as local image descriptors.” Ex. 1001, 7:36-40. “In general, the local descriptors may be of **any suitable type**,” and the '985 Patent provides “non-limitative examples,” such as “**values of individual pixels** or a linear combination thereof.” Ex. 1001, 9:54-58.

144. Like the '985 Patent, Ganin also discloses “values of individual pixels,” i.e., the “raw input pixels... [in] 3 color channels” (or RGB channels) of an input eye region. Ex. 1006, p. 6.

145. Ganin discloses “employ[ing] an external face alignment library” to produce “ $N = 7$ feature points $\{(x_i^{anchor}, y_i^{anchor}) \mid i = 1, \dots, N\}$ **for the eye** (six points along the edge and also the pupil center).” Ex. 1006, p. 5. Thus, in my opinion, these feature points (or anchor points) are “information about a local [eye] region” and therefore are “local descriptors.”

146. The '985 Patent also identifies “a difference between the pixels at two points” to be a local descriptor. Ex. 1001, 9:57-60. Similarly, Ganin also discloses “anchor maps,” where “for each... feature point located at $(x_i^{anchor}, y_i^{anchor})$, we compute a pair of maps:

$$\begin{aligned}\Delta_x^i[x, y] &= x - x_i^{\text{anchor}}, \\ \Delta_y^i[x, y] &= y - y_i^{\text{anchor}},\end{aligned}\quad \forall (x, y) \in \{0, \dots, W\} \times \{0, \dots, H\}, \quad (5)$$

where W, H are width and height of the input image [(e.g., input eye region)]

respectively.” Ex. 1006, p. 6. Here, $\Delta_x^i[x, y]$ and $\Delta_y^i[x, y]$ represent “difference[s]

between the pixels at two points” described in the ’985 Patent. Ex. 1001, 9:59-60.

Therefore, in my opinion, Ganin’s feature maps that include $\Delta_x^i[x, y]$ and $\Delta_y^i[x, y]$ are “local descriptors.”

147. Second, it is my opinion that, for each type of these local descriptors, i.e., pixel values of RGB channels, feature or anchor points, feature or anchor maps, Ganin discloses deriving a feature vector from them. As I discussed earlier in Section XII.A.1.b, during training, Ganin’s multi-scale NN takes three types of input data: (1) an input eye region, (2) feature points (anchors), and (3) a correction angle.

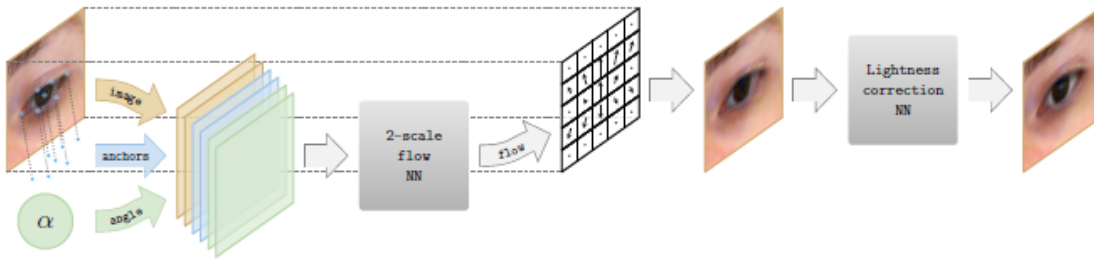


Fig. 2. The proposed system takes an input eye region, feature points (anchors) as well as a correction angle α and sends them to the multi-scale neural network (see Section 3.2) predicting a flow field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

Ex. 1006, Fig. 2 (colors original), Fig. 10(a). According to Ganin, “all inputs are expressed as maps,” which are concatenated together into an “input map stack” that is fed to Ganin’s multi-scale neural network. *Id.*, pp. 5-6. The “input map stack” uses the RGB channels as the initial map stack. *Id.*, p. 6. Ganin then discloses “[e]mbedding the [correction] angle” by deriving “a single position-independent 16-dimensional vector” from the correction angle, which “is then expressed as 16 constant maps that are concatenated into the input map stack.” *Id.* The input map stack at that point consists of 3 RGB maps and 16 angle embedding maps. Ganin further concatenates the feature or anchor maps mentioned above, forming “the input map stack consist[ing] of 33 maps ([3] RGB + 16 angle embedding maps + 14 feature point embedding maps).” *Id.*

148. In the process of forming the “input map stack,” it is my opinion that Ganin discloses deriving a feature vector in various stages. As a first example stage, when taking the pixel values of the RGB channels or the feature maps as local descriptors, it is my opinion that the input map stack is a concatenation of them and is therefore a feature vector derived therefrom.

149. A PHOSITA would have understood that in the field of image processing, the term “feature vector” does not necessarily refer to a one-dimensional vector; rather, a “feature vector” may refer to data with two-dimensions (such as an $M \times M$ map or a list of pairs of values) or even three

dimensions (such as a stack of maps). *See, e.g.*, Ex. 1046, 2:3-4 (“probability map is a first input vector”), 7:45-48 (“the matrix of the input feature vector”);

Alternatively, to the extent that Patent Owner argues that a feature vector must be a one-dimensional vector, it was well known to a PHOSITA that $M_1 \times M_2$ data are often transformed into $1 \times N$ vectors by scanning when provided to machine learning algorithms. For example, and consistent with a PHOSITA’s common general knowledge, Exhibit 1053 illustrates, for example in Figure 2, various ways of scanning “a two-dimensional array[] in a particular order to produce a one-dimensional vector.” Ex. 1053, 1:58-62.

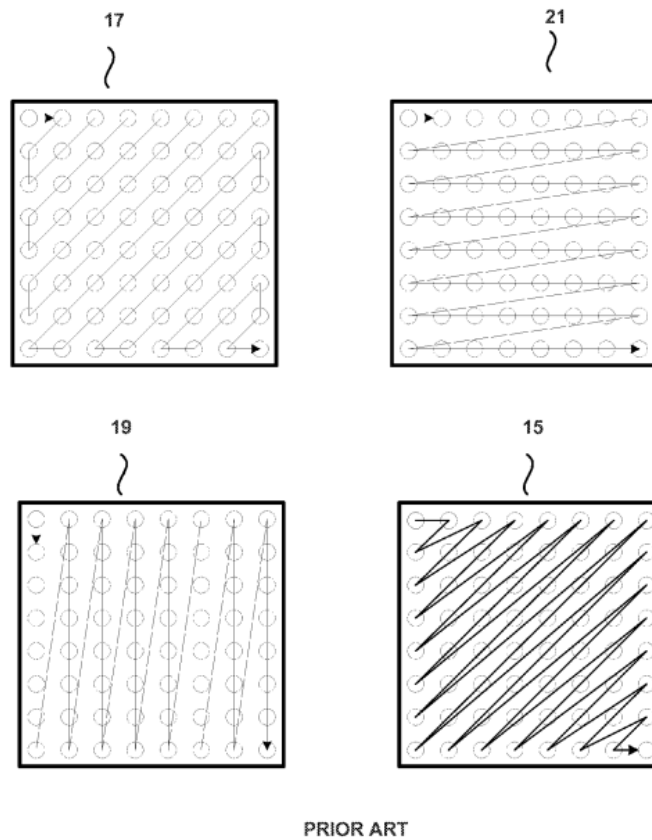


FIG. 2

Ex. 1053, Fig. 2; *see also* Ex. 1054, Figs. 5A-5B, 6A-6B, Figs. 8A-8B, and Fig. 9.

150. As a second example stage, also when taking the pixel values of the RGB channels as local descriptors, the feature or anchor points are extracted from these pixel values using “an external face alignment library,” and in my opinion, therefore constitute a feature vector derived therefrom. Ex. 1006, p. 5.

151. As a third example stage, Ganin discloses “comput[ing] a pair of [distance] maps” using anchor points $(x_i^{anchor}, y_i^{anchor})$:

$$\begin{aligned}\Delta_x^i[x, y] &= x - x_i^{\text{anchor}}, \\ \Delta_y^i[x, y] &= y - y_i^{\text{anchor}},\end{aligned}\quad \forall (x, y) \in \{0, \dots, W\} \times \{0, \dots, H\}, \quad (5)$$

where W, H are width and height of the input image respectively.” Ex. 1006, p. 6.

Thus, in my opinion, the “pair of [distance] maps” for an anchor point of an input image are local descriptors, and the combined pairs of distance maps for all the anchor points of the input image constitute a feature vector. Alternatively, it is my opinion that the anchor points themselves and corresponding pixel values may be considered local descriptors. Accordingly, in my opinion, either a pair of distance maps for an anchor point or the combined pairs of distance maps for all the anchor points constitute a feature vector.

152. Aside from Ganin’s on-point teachings of local descriptors, the ’985 Patent itself acknowledges that “local descriptors” were not new. The ’985 Patent incorporates by reference three prior-art publications teaching three types of local descriptors: “SIFT features,” “HOG features,” and “‘low level representations’ from pre-classification stages in deep learning neural networks.” Ex. 1001, 9:65-10:19 (citing “Distinctive Image Features from Scale-Invariant Keypoints” by Lowe, “Histograms of Oriented Gradients for Human Detection” by Dalal et al., “Multi-scale recognition with DAG-CNNs” by Yang and Ramanan). The ’985 Patent also acknowledges a paper titled “Learning To Look Up: Realtime Monocular Gaze Correction Using Machine Learning,” which uses “a distance of a

pixel location with the position of an eye landmark.” Ex. 1001, 10:61-67. The authors of the Kononenko paper are also listed co-authors of the Ganin paper and both papers disclose the same type of feature vector, i.e., “a distance of a pixel location with the position of an eye landmark.” *Id.*; Ex. 1006, p. 6; Ex. 1028, Fig. 2.

153. Thus, in addition to Ganin disclosing these features, it is my opinion that the admitted prior art in the '985 patent, additionally or alternatively, shows that the aspects recited in claim 2 were nothing new.

154. Third, it is also my opinion that Ganin discloses that “the first and second selection instructions define how the feature vector is used to select editing instructions for the input patch.” As I explained for claim 1 above, each of Ganin’s coarse and fine warping modules “takes as an input the image, the position of the feature points [embedded as feature maps], and the redirection angle [embedded as constant maps],” i.e., the “input map stack.” In my opinion, this is true for both runtime and training. Ex. 1006, p. 6. The trained coarse warping model takes the “input image stack” and generates the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$. Similarly, the trained fine warping model also takes the “input image stack” and other parameters as appropriate, and generates “residual flow” $\mathbf{D}_{\text{res}}$ and then final warping field $\mathbf{D}(I, \alpha)$.

155. Therefore, in my opinion, Ganin discloses that “the configuration of the target feature of each input patch is represented by a feature vector [e.g., “input map stack”] derived from plural local descriptors [e.g., $N=7$ feature points and pixel values within input patch] of the input patch, and the first and second selection instructions [e.g., trained coarse and fine warping modules] define how the feature vector is used to select editing instructions [e.g., coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ for coarse warping module, and “residual flow” $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ for fine warping module] for the input patch.”

3. Claim 3

156. In my opinion, claim 3 is unpatentable because it is anticipated by Ganin. Claim 3 of the '985 Patent depends from claim 1 and recites:

3. The method of claim 1, wherein the editing instructions comprise a displacement vector field defining how the input patch is to be transformed.

157. In my opinion, Ganin discloses the added feature of “wherein the editing instructions comprise a displacement vector field defining how the input patch is to be transformed” by teaching using a “flow field,” or a “warping field” to transform an input eye region into a gaze-corrected eye region. Like the '985 Patent, Ganin transforms pixels of the input image I using coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$, resulting in the rough image estimate (i.e., coarse gaze correction):

$$O_{\text{coarse}}(x, y, c) = I\{x + \mathbf{D}_{\text{coarse}}(I, \alpha)(x, y, 1), y + \mathbf{D}_{\text{coarse}}(I, \alpha)(x, y, 2), c\}, \quad (2)$$

Ex. 1006, p.5, Eq.2. Similarly, Ganin also transforms pixels of the input image I using final warping field $\mathbf{D}(I, \alpha)$, resulting in the final estimate (i.e., fine gaze correction):

$$O = S(I, \mathbf{D}(I, \alpha)) . \quad (4)$$

Ex. 1006, p.6, Eq.4. The '985 Patent specifically acknowledges that its “displacement vector fields” may be “referred to as a ‘flow field’” (Ex. 1001, 9:11-12), which is another way Ganin refers to its “warping fields.” Ex. 1006, pp. 4, 18, Fig. 2, Fig. 10.

158. Therefore, in my opinion, Ganin discloses that “the editing instructions comprise a displacement vector field [e.g., “flow field” or “warping field”] defining how the input patch is to be transformed [e.g., gaze correction by warping pixels].”

159. In addition to Ganin disclosing the '985 Patent's “displacement vector field,” the '985 Patent also states:

“By way of non-limitative example, **the displacement vector field F may be derived as disclosed in Kononenko et al., “Learning To Look Up: Realtime Monocular Gaze Correction Using Machine Learning”, Computer Vision and Pattern Recognition,**

2015, pp. 4667-4675, which is herein incorporated by

reference in its entirety, wherein **the displacement**

vector field F is referred to as a “flow field”.”

Ex. 1001, 9:6-12 (emphasis added). So in my opinion, this added feature for claim 3 adds nothing novel to claim 1, since the '985 Patent admits that this feature is part of the prior art.

4. Claim 5

160. In my opinion, claim 5 is unpatentable because it is anticipated by Ganin. Claim 5 of the '985 Patent recites the following. I address each of these in my analysis below.

5[P] The method of claim 1, wherein:

5[A] a first editing algorithm is used by the first machine learning algorithm to define how the first editing instructions are to be applied to an input patch to derive an output patch; and

5[B] a second editing algorithm is used by the second machine learning algorithm to define how the second editing instructions are to be applied to an input patch to derive an output patch.

a. 5[P] and 5[A]

161. As I explained above, Ganin discloses claim 1. In my opinion, Ganin also discloses 5[A].

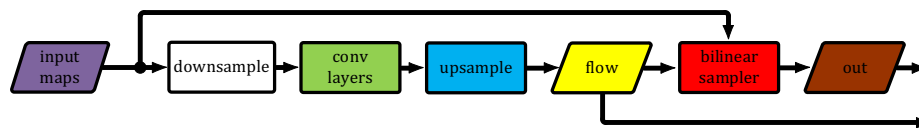
162. As I explained above, Ganin discloses “first editing instructions,” e.g., coarse warping field $D_{\text{coarse}}(I, \alpha)$. Ganin further discloses that “ $D_{\text{coarse}}(I, \alpha)$ [is] applied to **warp the input image** by means of a **bilinear sampler S** that finds the *coarse estimate*:

$$O_{\text{coarse}} = S(I, D_{\text{coarse}}(I, \alpha)) . \quad (1)$$

Ex. 1006, p. 5, Eq. 1. “Here, the sampling procedure S samples the pixels of O_{coarse} at pixels determined by the flow field:

$$O_{\text{coarse}}(x, y, c) = I\{x + D_{\text{coarse}}(I, \alpha)(x, y, 1), y + D_{\text{coarse}}(I, \alpha)(x, y, 2), c\} , \quad (2)$$

where c corresponds to a color channel (R, G, or B), and the curly brackets correspond to bilinear interpolation of $I(\cdot, \cdot, c)$ at a real-valued position. The sampling procedure (1) is piecewise differentiable.” *Id.* Ganin also illustrates this process in Fig. 10(b) below, where the bilinear sampler (red) is used to apply the coarse warping fields (yellow) to input images (represented by input maps (purple)) to generate rough gaze-corrected image estimates.



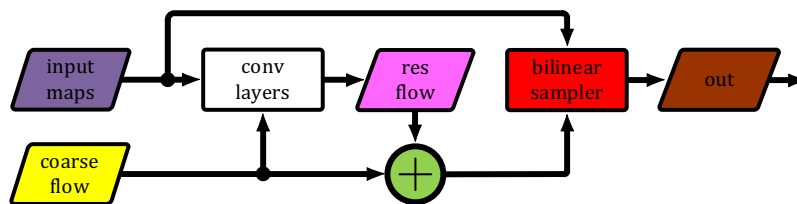
(b) 0.5×-scale processing module

Ex. 1006, Fig. 10(b).

163. Therefore, in my opinion, Ganin discloses “a first editing algorithm [e.g., bilinear sampler S] is used by the first machine learning algorithm [e.g., coarse warping module] to define how the first editing instructions [e.g., coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$] are to be applied to an input patch [e.g., “input image”] to derive an output patch [e.g., rough image estimates or “coarse estimate”].”

b. 5[B]

164. Ganin’s fine warping module generates a residual flow $\mathbf{D}_{\text{res}}$, which is added to the coarse warping field $\mathbf{D}_{\text{coarse}}(I, \alpha)$ to obtain the final warping field $\mathbf{D}(I, \alpha)$. Ex. 1006, pp. 5-6. Like its coarse warping module, Ganin also uses a bilinear sampler to apply the final warping field $\mathbf{D}(I, \alpha)$ to warp the input image I “to obtain the final output... $O = \mathbf{S}(I, \mathbf{D}(I, \alpha))$.” *Id.*, p. 6, Eq. 4. Ganin also illustrates this process in Fig. 10(c) below, where the bilinear sampler (red) is used to apply the combined residual flow (pink) and coarse warping field (yellow) to input images (represented by input maps (purple)) to generate final gaze-corrected images (brown).



Ex. 1006, Fig. 10(c).

165. Therefore, in my opinion, Ganin discloses “a second editing algorithm [e.g., bilinear sampler S] is used by the second machine learning algorithm [e.g., fine warping module] to define how the second editing instructions [e.g., residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$] are to be applied to an input patch [e.g., “input image”] to derive an output patch [e.g., final gaze corrected images].”

5. Claim 10

166. In my opinion, claim 10 is unpatentable because it is anticipated by Ganin. Claim 10 of the '985 Patent depends from claim 1 and recites:

10. The method of claim 1, wherein the first machine learning algorithm comprises one or more of the following: **a neural network**; a support vector machine; and a generative adversarial network (GAN).

167. As I explained in Section XII.A.1.e, Ganin discloses that “the first machine learning algorithm comprises... a neural network [e.g., 0.5×-scale NN]”. In my opinion, claim 10 is therefore anticipated.

6. Claim 12

168. In my opinion, claim 10 is unpatentable because it is anticipated by Ganin. Claim 10 of the '985 Patent depends from claim 1 and recites:

12. The method of claim 1, wherein the target feature comprises one or more of the following: **an eye region comprising at least part of an eye**, a nose region comprising at least part of a nose, a mouth region comprising at least part of a mouth, a chin region comprising at least part of a chin, a neck region

comprising at least part of a neck, and a hair region comprising hair.

169. As I explained in Sections XII.A.1.c-XII.A.1.d, Ganin discloses that “target feature comprises... an eye region comprising at least part of an eye.” In my opinion, claim 12 is therefore anticipated.

7. Claim 13

170. In my opinion, claim 13 is unpatentable because it is anticipated by Ganin. Claim 13 of the '985 Patent recites the following. I address each of these in my analysis below.

13[P] A method of adjusting a digital representation of a head region, the method comprising

13[A] receiving a digital representation of a head region; and

13[B] using reference data comprising editing instructions to adjust the digital representation of the head region, wherein the reference data comprises the second reference data generated by the method of claim 1.

a. 13[P]

171. Ganin proposes an improved “method” for “gaze manipulation.” Ex. 1006, p. 1. As shown in Figure 2 below, Ganin’s method uses a “multi-scale neural network [to] predict[] a flow field,” which “is then applied to the input image to produce an image of a redirected eye.” *Id.*, p. 4; *see also supra* Section XII.A.1.

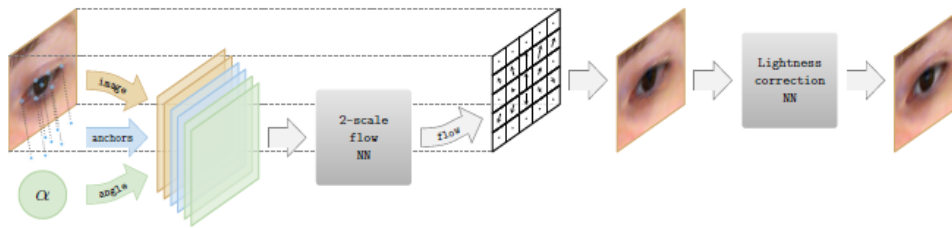


Fig. 2. The proposed system takes an input eye region, feature points (anchors) as well as a correction angle α and sends them to the multi-scale neural network (see Section 3.2) predicting a flow field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

Ex. 1006, Fig. 2 (colors original).

172. Therefore, to the extent 13[P] is limiting, it is my opinion that Ganin discloses “a method of adjusting a digital representation of a head region [e.g., gaze redirection for an input eye region].”

b. 13[A]

173. As shown in Fig. 2 below, the input to Ganin’s gaze redirection method includes “an **input eye region**, feature points (anchors) as well as a correction angle α .”

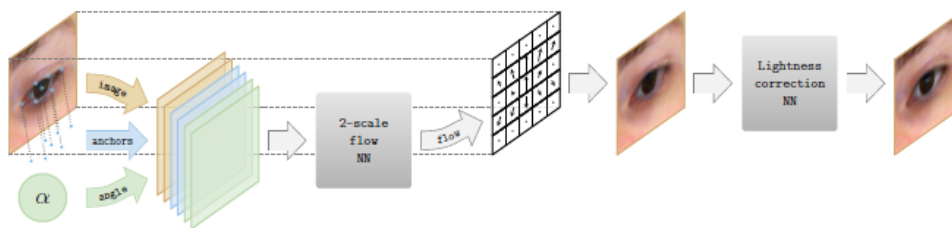


Fig. 2. The proposed system takes an input eye region, feature points (anchors) as well as a correction angle α and sends them to the multi-scale neural network (see Section 3.2) predicting a flow field. The flow field is then applied to the input image to produce an image of a redirected eye. Finally, the output is enhanced by processing with the lightness correction neural network (see Section 3.4).

Ex. 1006, Fig. 2; *supra* Section XII.A.2 (explaining how the input eye region is

combined with two other parameters to form an “input map stack” to be received by the multi-scale neural network).

174. Therefore, in my opinion, Ganin discloses “receiving a digital representation of a head region [e.g., an input eye region].”

c. 13[B]

175. As I explained above, it is my opinion that Ganin discloses generating “second reference data” **during training**, which includes “second selection instructions” (e.g., trained $1\times$ -scale NN) and “second editing instructions,” (e.g., residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$). *Supra* Section XII.A.1.h.

176. In my opinion, Ganin also discloses using the “second reference data” generated by the method of claim 1, i.e., generated **during training**, for gaze redirection **during runtime**.

177. First, as I explained above, Ganin uses the trained $1\times$ -scale NN to generate residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ during runtime. *Supra* Sections XII.A.1.g-XII.A.1.h. Thus, in my opinion, by using the “second selection instructions” (e.g., the trained $1\times$ -scale NN, which is part of the “second reference data” generated during training), Ganin discloses using the “second reference data” for runtime gaze redirection.

178. Second, residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ are used in Ganin for gaze redirection and are generated **during both training and runtime**.

See Supra Section XII.A.1.f. The residual flows $\mathbf{D}_{\text{res}}$ and final warping fields $\mathbf{D}(I, \alpha)$ generated during training encompass all the possible values of residual flow $\mathbf{D}_{\text{res}}$ and/or final warping field $\mathbf{D}(I, \alpha)$. During runtime, the trained $1\times$ -scale NN essentially picks, for the input image, an optimal residual flow $\mathbf{D}_{\text{res}}$ and therefore optimal final warping field $\mathbf{D}(I, \alpha)$ from the possible values of residual flows $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ generated during training, or from a representation of these through the warping-fields generating function that is learned by the network during the training phase. Thus, in my opinion, by picking the optimal residual flow $\mathbf{D}_{\text{res}}$ and final warping field $\mathbf{D}(I, \alpha)$ from the “second editing instructions” (e.g., the residual flows $\mathbf{D}_{\text{res}}$ and final warping fields $\mathbf{D}(I, \alpha)$ generated during training or representation thereof, which are part of the “second reference data”), Ganin discloses using the “second reference data” for runtime gaze redirection.

179. In my opinion, claim 13 is therefore anticipated.

8. Claim 14

180. In my opinion, claim 14 is unpatentable because it is anticipated by Ganin. Claim 14 of the '985 Patent recites the following:

14. A non-transitory computer readable storage medium storing a computer program capable of execution by a processor and arranged on execution to cause the processor to perform a method according to claim 1.

181. As I explained above, Ganin discloses a method according to claim 1. *Supra* Section XII.A.1. Ganin further discloses “computational performance of our method” “on a mid-range consumer GPU (NVIDIA GeForce-750M),” and also contemplates performance of its method “on CPU.” Ex. 1006, p. 14. “GPU” and “CPU” stand for “Graphical Processing Unit” and “Central Processing Unit” respectively, both of which are a type of “processor.”

182. It is also my opinion that a PHOSITA would have understood that in order for a processor to perform such sophisticated processes involving image processing and machine learning, such processor necessarily needs to execute a computer program containing instructions for each step recited in claim 1, which is necessarily stored on a non-transitory computer readable medium. *See, e.g.*, Ex. 1029 (Github page: Tensorflow implementation of Ganin); Ex. 1007, Fig. 8, ¶74. In fact, that is one main aspect of “machine learning.”

183. Thus, in my opinion, claim 14 is anticipated.

9. Claim 18

184. In my opinion, claim 18 is unpatentable because it is anticipated by Ganin. Claim 18 of the '985 Patent recites the following. I address each of these in my analysis below.

18[P] A method of generating reference data for adjusting a digital representation of a head region, the method comprising:

18[A] receiving training data comprising:

18[B] a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

18[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

18[D] using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and

18[E] using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

18[F] wherein the target feature comprises an eye region comprising at least part of an eye and the adjustment of the digital representation of the head region comprises adjusting a gaze direction, and/or

wherein the target feature comprises a nose region comprising at least part of a nose and the adjustment of the digital representation of the head

region comprises adjusting a shape and/or texture of the nose, and/or

wherein the target feature comprises a chin region comprising at least part of a chin and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the chin, and/or

wherein the target feature comprises a neck region comprising at least part of a neck and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the neck, and/or

wherein the target feature comprises a hair region comprising hair and the adjustment of the digital representation of the head region comprises adjusting a color of the hair.

a. 18[P] and 18[A-E]

185. As I show below, 18[P] and 18[A-E] are identical to 1[P] and 1[A-E], respectively.

Claim 18 Limitation		Claim 1 Limitation	
18[P]	A method of generating reference data for adjusting a digital representation of a head region, the method comprising:	1[P]	A method of generating reference data for adjusting a digital representation of a head region, the method comprising:
18[A]	receiving training data comprising:	1[A]	receiving training data comprising:
18[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region	1[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region

	prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and		prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
18[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;	1[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
18[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and	1[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
18[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital	1[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital

	representation of the head region for a range of possible digital representations of the head region,		representation of the head region for a range of possible digital representations of the head region,
--	---	--	---

186. Therefore, in my opinion, and 18[P] and 18[A-E] are disclosed for the same reasons I provided for in Sections XII.A.1.a-XII.A.1.f.

b. 18[F]

187. 18[F] is substantially similar to an aspect in Claim 12: (identical portion indicated in bold)

Claim 18 Limitation		Claim 2 Limitation	
18[F]	wherein the target feature comprises an eye region comprising at least part of an eye and the adjustment of the digital representation of the head region comprises adjusting a gaze direction, and/or wherein the target feature comprises a nose region comprising at least part of a nose and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the nose, and/or wherein the target feature comprises a chin region comprising at least part of a chin and the adjustment of the	12	12. The method of claim 1, wherein the target feature comprises one or more of the following: an eye region comprising at least part of an eye, a nose region comprising at least part of a nose, a mouth region comprising at least part of a mouth, a chin region comprising at least part of a chin, a neck region comprising at least part of a neck, and a hair region comprising hair.

	digital representation of the head region comprises adjusting a shape and/or texture of the chin, and/or wherein the target feature comprises a neck region comprising at least part of a neck and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the neck, and/or wherein the target feature comprises a hair region comprising hair and the adjustment of the digital representation of the head region comprises adjusting a color of the hair.		
--	--	--	--

188. Therefore, in my opinion, for the same reasons I provided in Section XII.A.6, 18[F] is also taught by Ganin.

B. My Opinions as to Claims 16 and 19 based on Ganin in view of Liu.

1. Claim 16

189. In my opinion, claim 16 is unpatentable because it is rendered obvious by Ganin in view of Liu. Claim 16 of the '985 Patent recites the following. I address each of these in my analysis below.

16[P] A device for generating reference data for adjusting a digital representation of a head region, the device comprising a data processing unit arranged to:

16[A] receive training data comprising:

16[B] a set of input patches, each input patch comprising information about a target feature of a digital representation of the head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

16[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target portion of the digital representation of the head region after adjustment of the digital representation of the head region;

16[D] use a first machine learning algorithm to generate first reference data using the training data; and

16[E] use a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data output by the first machine learning algorithm

16[F] wherein the first reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

16[G] wherein the second reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region, and

16[H] wherein the second editing instructions in the second reference data are provided in a compressed representation.

a. 16[P]

190. As shown below, most of 16[P] is identical to 1[P] (identical portion indicated in bold).

Claim 16 Limitation		Claim 1 Limitation	
16[P]	A device for generating reference data for adjusting a digital representation of a head region , the device comprising a data processing unit arranged to:	1[P]	A method of generating reference data for adjusting a digital representation of a head region , the method comprising:

191. Ganin also discloses that its method may be performed on a “mid-range consumer GPU (NVIDIA GeForce-750M)” or “CPU.” Ex. 1006, p. 14. GPU stands for “Graphics Processing Unit” and CPU stands for “Central Processing Unit.” Both GPU and CPU are “data processing unit[s]” required by 16[P] and in my opinion, the device containing such GPU or CPU is the recited device in claim 16.

192. Thus, in my opinion, 16[P] is disclosed in Ganin for the same reasons I explained in Section XII.A.1.a above.

b. 16[A-G]

193. As I show below, 16[A-G] collectively are substantively identical to 1[A-E] collectively.

Claim 16 Limitation		Claim 1 Limitation	
16[A]	receive training data comprising:	1[A]	receiving training data comprising:
16[B]	a set of input patches, each input patch comprising information about a target feature of a digital representation of the head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and	1[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
16[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target portion of the digital representation of the head region after adjustment of the digital representation of the head region;	1[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
16[D]	use a first machine learning algorithm to generate first reference data using the training data; and	1[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
16[F]	wherein the first reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,		

16[E]	use a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data output by the first machine learning algorithm	1[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,
16[G]	wherein the second reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region, and		

194. Thus, in my opinion, 16[A-G] are disclosed in Ganin for the same reasons I explained in Sections XII.A.1.b-XII.A.1.f above.

195. I note that 16[B] recites “information about a target feature,” as opposed to just “a target feature” in 1[B]. However, this does not change the scope of this limitation and does not affect my analysis.

196. I also note that 16[C] uses the term “target portion,” as opposed to “target feature” in 1[C]. This appears to be a typographical error and does not affect my analysis. *See* Ex. 1001, 1:67-2:4

c. 16[H]

197. 16[H] recites that “the second editing instructions in the second reference data are provided in a compressed representation,” which, in my opinion, is disclosed by Ganin in view of Liu.

198. According to the '985 Patent, a compressed representation may be a principal component analysis representation. *See* Ex. 1001, 15:59-61. Liu discloses applying “PCA... to the two-dimensional (2D) data of optical flows.” Ex. 1016, p.1. Ganin’s warping fields correspond to pixel-flow fields that provide displacements or motions to pixels and in my opinion, are a type of optical flows. As I will explain in Section XII.B.2 below, a PHOSITA would have been motivated to use Liu’s PCA to compress Ganin’s warping fields, i.e., second editing instructions.

199. Therefore, in my opinion, Ganin in view of Liu discloses that “the second editing instructions in the second reference data are provided in a compressed representation [e.g., PCA representation].”

2. Motivation to Combine Ganin with Liu

200. In my opinion, a PHOSITA would have been motivated to apply Liu’s PCA to Ganin’s warping fields.

201. Ganin and Liu qualify as analogous art to the '985 Patent because they are in the same field of endeavor. Ex. 1006, Fig. 2; Ex. 1016, p. 1 (Introduction);

Ex. 1001, 9:6-12. For example, both references and the '985 Patent concern pixel flow fields. Liu relates to analyzing optical flows that have happened and reflected in the pixels of images. Ex. 1016, p. 1. Both Ganin and the '985 Patent relate to analyzing motion vector fields to be used to conduct gaze correction. At a minimum, Liu is at least reasonably pertinent to a problem identified in the '985 Patent. Ex. 1001, 15:57-61; Ex. 1016, p. 1.

202. It is my opinion that a PHOSITA would have been motivated to apply Liu's PCA to Ganin's warping fields to achieve the advantages disclosed in Liu, and Ganin itself encourages further improvements in its system. Specifically, Liu discloses that applying PCA to optical flows involves "project[ing] the original data onto an orthogonal subspace" and using "first several principal components" to represent "major data information." Ex. 1016, p. 1. This can "reduce false alarms for videos with either static or dynamic background, and it is also useful to delineate the size of moving objects." *Id.* Liu further discloses that "if PCA is applied to the two-dimensional (2D) data of optical flows, the difference between the desired motion pixels and random motion pixels [e.g., due to noise] can be magnified," which can help in filtering out the motion component due to noise. *Id.* Ganin also encourages further improvements to its system (Ex. 1006); so a PHOSITA would have naturally looked to an analogous reference like Liu to further improve Ganin's system. For example, Ganin discloses that "the residual

flow predicted by the 1x-scale module tend to be quite noisy.” Ex. 1006, p.8. A PHOSITA would thus have been motivated to look for and apply techniques such as Liu’s PCA to minimize the noise in Ganin’s predicted residual flow. When applied to Ganin’s warping fields, Liu’s PCA compresses Ganin’s warping fields such that Ganin’s major warping information is represented by PCA components, thereby reducing unexpected pixel warping, possibly caused by noise, movement, or the person’s facial appearance.

203. Finally, a PHOSITA would have had a reasonable expectation of success in applying Liu’s PCA to Ganin’s warping fields. As Liu recognizes, “[p]rinciple component analysis (PCA) is a typical approach in multivariate analysis.” Ex. 1016, p.1. Not only does Liu provide a detailed description of its PCA approach, PCA was also well known before the ’985 patent and widely used in image processing. *See, e.g.*, Ex. 1016, p. 1 (“Principal component analysis (PCA) is a typical approach in multivariate analysis.”) (citing K. Pearson, “On lines and planes of closest fit to systems of points in space,” *Philosophical Magazine*, vol. 2, no. 6, pp. 559–572, 1901.); Ex. 1030, pp. 56-57. (I noted that the provided content in Exhibit 1030 is an enhanced version uploaded by its author on 14 May 2014.) Thus, it is my opinion that applying PCA to Ganin’s warping fields was a well-known process to implement and would have resulted in a working system.

204. Therefore, in my opinion, Ganin-Liu renders obvious claim 16.

3. Claim 19

205. In my opinion, claim 19 is unpatentable because it is rendered obvious by Ganin in view of Liu. Claim 19 of the '985 Patent recites the following. I address each of these in my analysis below.

19[P] A method of generating reference data for adjusting a digital representation of a head region, the method comprising:

19[A] receiving training data comprising:

19[B] a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

19[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

19[D] using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and

19[E] using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second

reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

19[F] wherein the second editing instructions in the second reference data are provided in a compressed representation, and

19[G] wherein the compressed representation comprises one or more of the following: a principle component analysis representation; a wavelet representation; Fourier and/or discrete cosine transform components; cluster centers.

a. 19[P] and 19[A-E]

206. As I show below, 19[P] and 19[A-E] are identical to 1[P] and 1[A-E], respectively.

Claim 19 Limitation		Claim 1 Limitation	
19[P]	19. A method of generating reference data for adjusting a digital representation of a head region, the method comprising:	1[P]	A method of generating reference data for adjusting a digital representation of a head region, the method comprising:
19[A]	receiving training data comprising:	1[A]	receiving training data comprising:
19[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each	1[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each

	input patch; and		input patch; and
19[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;	1[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
19[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and	1[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
19[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,	1[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

207. Therefore, in my opinion, 19[P] and 19[A-E] are disclosed by Ganin for the same reasons I provided in Sections XII.A.1.a-XII.A.1.f.

b. 19[F-G]

208. Ganin-Liu includes having the second editing instructions provided in a PCA representation, and, because PCA is a form of compression, it is my opinion that the additional aspects in 19[F-G] are rendered obvious by Ganin in view of Liu for the same reasons I explained in Section XII.B.1.c.

C. **My Opinions as to Claims 1-3, 5-6, 8-14, and 16-19 in view of the Chalom-Aslan Combination**

1. **Motivation to Combine Chalom and Aslan**

209. As I will explain for each limitation below, it is my opinion that Chalom discloses all the features in claim 1 except for the training of two MLAs, which is disclosed by Aslan's teacher-student architecture for training. It is also my opinion that ample motivations existed to incorporate Aslan's teacher-student architecture to train Chalom's random forest classifier ("RFC").

210. It is my opinion that Chalom and Aslan qualify as **analogous art** at least because they are in the **same field of endeavor** as the '985 Patent, *i.e.*, using machine learning for image processing. Both references (and the '985 Patent) disclose using training data, such as images, to train MLAs. Ex. 1001, Abstract; Ex. 1005, 3:21-32; Ex. 1007, ¶24. Specifically, both references (and the '985 Patent) teach extracting features from input images (for training and for runtime)

and providing these extracted features to the MLAs. Ex. 1001, Figs. 3, 8 and 11; Ex. 1005, Figs. 3, 8, 13 and 14; Ex. 1007, ¶24.

211. Like the '985 Patent and Chalom, Aslan also discloses that “features included in the training data 104 can be represented... in the form of an **n-dimensional feature vector** of quantifiable information about an attribute of the training data 104.” Ex. 1007, ¶24; Ex. 1001, 7:36-40; Ex. 1005, 8:26-30. “For example, if the training data 104 comprises a repository of images, the **feature vector** can include values that correspond to the **pixels of the image**, the size (length, height, area, etc.) and/or shape of objects, color, hue, saturation, and/or intensity, and so on.” Ex. 1007, ¶24.

212. At a minimum, it is my opinion that Chalom and Aslan are at least reasonably pertinent to a problem identified in the '985 Patent. For example, the '985 Patent indicates that high accuracy MLAs often consume more computational resources and are slower to run, as compared to efficient MLAs tending to provide less accurate predictions. Ex. 1001, 16:24-30; Ex. 1007, ¶45. This is not a new issue discovered by the '985 Patent inventors. *See* Section V.D.3. Chalom and Aslan are at least reasonably pertinent to this efficiency-versus-accuracy problem. Chalom teaches a machine-learning system that is not only “fast enough to support real time implementation” but also “robust enough in terms of quality,” e.g., by “reducing the histogram of oriented gradient features” provided to the Random

Forest Classifier (RFC) 306 (yellow) and predicting “motion vector field[s]
[green]... to warp the eye region [blue].” Ex. 1005, 1:16-25, 9:48-55, 10:40-60.

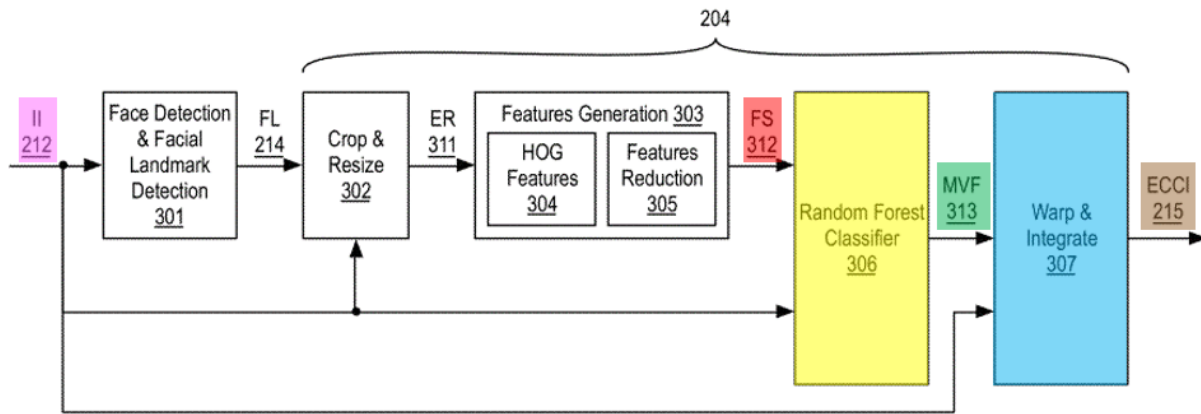


FIG. 3

Ex. 1005, Fig. 3. Similarly, Aslan (and the '985 Patent) adopts a teacher-student architecture in the training process such that the efficient student model learns from the accurate teacher model, thereby maintaining the efficiency of the student model while improving its accuracy. Ex. 1007, ¶45; Ex. 1001, 16:24-44, 17:14-17.

213. In my opinion, a PHOSITA also would have been motivated to incorporate Aslan’s teacher-student architecture into Chalom’s training of its RFC 306 for accuracy and/or efficiency gains.

214. First, Chalom contemplates that its RFC 306 “may **apply different random forest models** responsive to or selectively based on camera 104 and display 102 having a particular relative position therebetween.” Ex. 1005, 10:36-39. In exploring ways to implement the use of different random forest models to

construct the systems described by Chalom, it is my opinion that a PHOSITA would have naturally consulted analogous references like Aslan and would have learned of Aslan's teacher-student model architecture being applied to image processing.

215. While Aslan describes the teacher-student architecture in the image-classification context, in my opinion, a PHOSITA would have been motivated to implement similar techniques for Chalom's image-editing context. This is at least because determining, using machine learning, an optimal motion vector field for a given eye region is still an image-classification task. Chalom's training process teaches all the possible motion vector fields to its MLAs, and these possible motion vector fields, although they will be used for image editing, they are classes/labels themselves. During runtime, finding the optimal motion vector field corresponds to finding the best class/label and therefore involves a classification process. *See also, e.g.,* Sections V.D.2 and V.F.

216. Alternatively, or in addition, Chalom expressly motivates looking for ways to further compress its random forest by indicating that it "may be compressed using any suitable technique or techniques." Ex. 1005, 15:4-7. In my opinion, this would have alternatively or additionally motivated a PHOSITA to look for such "suitable technique or techniques," and such PHOSITA would have been led to Aslan and its teachings related to "model compression" that allow for

creating a second (student) model that “can comprise a much smaller machine learning model...” Ex. 1007, ¶44.

217. Thus, in my opinion, upon learning of Aslan’s teacher-student model architecture and its application to image processing, a PHOSITA would have been motivated to apply that architecture in training the different random forest models described by Chalom, and would have done so to achieve Aslan’s gains in efficiency and accuracy for the student machine learning model. Ex. 1007, ¶44.

218. Second, it is my opinion that a PHOSITA would have been motivated to incorporate Aslan’s teacher-student architecture into Chalom’s system in different ways.

219. One way is using Chalom’s RFC as a teacher model to train an even more compact MLA, e.g., a smaller random forest.

220. It is my opinion that a PHOSITA would have understood that Chalom’s RFC is a machine-learning ensemble with a collection of decision trees. A machine-learning ensemble with a large number of decision trees or with relatively complex decision trees (more levels and larger number of nodes) requires more time and resources to run as compared to a random forest with a smaller number of decision trees and/or with less complex decision trees (e.g., fewer number of levels/nodes per tree). Aslan teaches that both the teacher and student machine learning models can be “tree-based models” and/or “an

ensemble”. EX 1007, ¶22. Aslan also explains that “[o]ften, the best performing machine learning models – in terms of the accuracy of the model’s output” are ensemble models comprising a large number of “base-level machine learning models” and that “because [such large] ensembles typically require a relatively **large storage footprint and powerful processing resources to execute at runtime**, they are not well suited for implementations where storage space and/or computational power is at a premium (such as with smart phones, wearables, hearing aids, etc.).” *Id.*, ¶3.

221. Aslan therefore adopts an improved teacher-student architecture to solve this problem. For example, Aslan explains that “the smaller second model 102 take much less memory to maintain and can operate faster on less processing power at runtime, the second model 102 can be a compressed form of the larger first model 100 such that the second model 102 can be more readily deployed on computing devices with limited resources (e.g., mobile devices, wearables, etc.).” Ex. 1007, ¶44.

222. Thus, in my opinion, a PHOSITA would have been motivated to incorporate Aslan’s teacher-student training architecture by **using Chalom’s already trained RFC as a teacher** to train a more compact RFC with fewer and/or less complex decision trees, thereby maintaining accuracy of the original RFC while achieving efficiency gains.

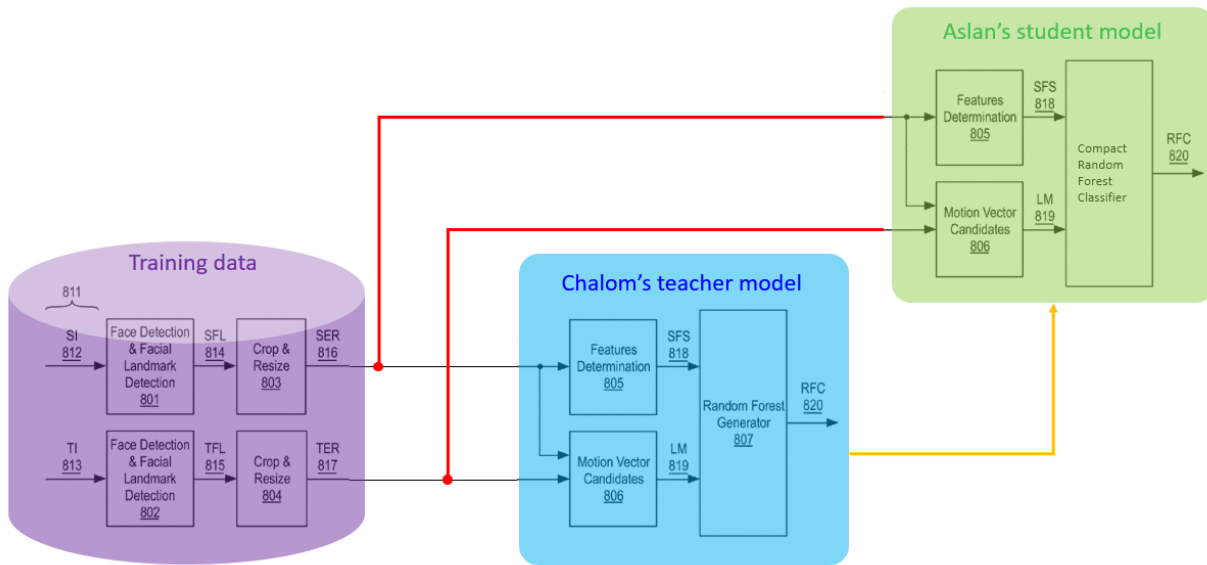
223. Another way a PHOSITA would have been motivated to incorporate Aslan's teacher-student architecture is by using Chalom's RFC as a student model, which learns from a different teacher model.

224. As recognized by the '985 Patent, it was well known at the time of the '985 Patent that using a random forest resulted in less optimal prediction accuracy as compared to deep convolutional neural networks for image processing/classification tasks. *See, e.g.*, Ex. 1001, 16:24-30. In my opinion, a PHOSITA looking to improve the accuracy for Chalom's RFC would have been motivated to incorporate Aslan's teacher-student training architecture by **training a new accurate teacher model first, which is then used to train Chalom's RFC as a student model**. As taught by Aslan, such new teacher model could be, e.g., a deep neural network or another type of machine learning model. Ex. 1007, ¶45. Thus, it is my opinion that a PHOSITA would have been motivated to train Chalom's RFC 306 by taking advantage of the teacher-student architecture provided by Aslan and achieving those advantages identified in Aslan from using a teacher-student architecture.

225. Finally, it is my opinion that a PHOSITA would have had a **reasonable expectation of success** in incorporating Aslan's teacher-student training architecture into Chalom's system.

226. For example, Aslan discloses using the same training data to train both the teacher and the student (Ex. 1007, Fig. 1) and thus no modification is needed with respect to training data preparation. Both Chalom and Aslan disclose using features derived from images to feed the MLA (Ex. 1005, 13:41-52; Ex. 1007, ¶24), and thus, in my opinion, no modification is needed with respect to training data pre-processing, i.e., the HOG feature generation and Principal Component Analysis thereof.

227. When using Chalom's RFC as a teacher to train a more compact student, it is my opinion that a PHOSITA would have needed only to follow the instructions provided in Aslan, such that after training the RFC (blue, below) the exact way Chalom does, feed the same training data (purple) and the output of Chalom's random forest (orange) to train a student model, e.g., a smaller or more compact random forest (green). In my opinion, such training of a random forest was a well-known process (as acknowledged by the '985 Patent, Ex. 1001, 11:1-6), and would have been an easy-to-achieve repetition of the teacher's training process. An example illustration of the Chalom-Aslan combination is shown below.



Ex. 1005, Fig. 8, modified based on Aslan to incorporate a student model; *c.f.* Ex. 1007, Fig. 1. As shown, the same training data, e.g., source eye regions 816 and target eye regions 817 are provided to both the teacher model and the student model. Further optimizations and predictable variations also would have been apparent to a PHOSITA, including, e.g., (i) having one features determination module 805 and one motion vector candidates module 806 to provide source feature sets 818 and likelihood maps 819 to both the teacher and student models; and (ii) using another type of MLA (other than a RFC) as the student model. *See* Ex. 1005, 3:27 (“a pretrained classifier such as a random forest”); 9:14-16 (“any suitable pretrained classifier may be applied such as, for example, a decision tree learning model, a kernel random forest model, or the like.”).

228. I understand that as shown in Aslan's Figure 1 below, in addition to the red arrow 106 passing information from the teacher model to the student model, Aslan also includes two orange arrows, i.e., 108 and 110.

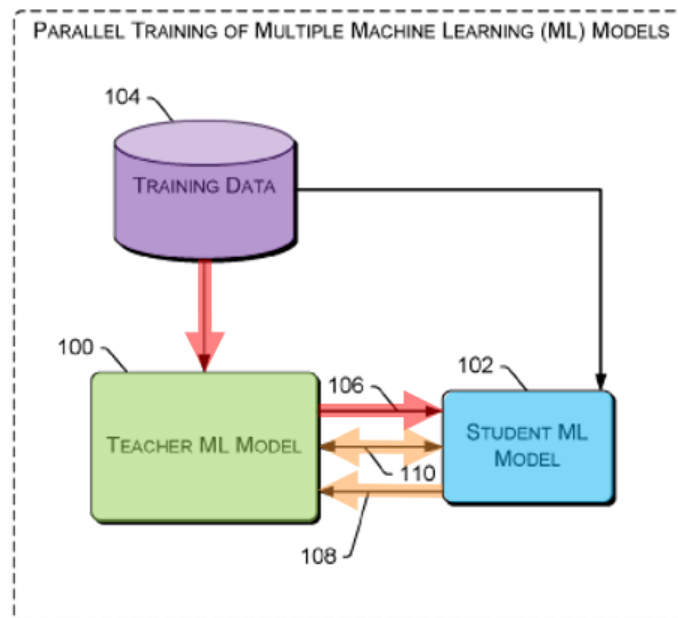


Fig. 1

Ex. 1007, Fig. 1 (annotated by adding colors). According to Aslan, “[i]n some implementations, the second model 102 is trained in parallel with the training of the first model 100 by providing some or all of the training data 104 to the second model 102, as depicted visually in FIG. 1 by the path 108 going from the training data 104 to the second model 102, and from the second model 102 to the first model 100.” Ex. 1007, ¶29. In addition, “knowledge can be bi-directionally transferred between the first model 100 and the second model 102 during joint

training, as depicted visually in FIG. 1 by path 110 between the first model 100 and the second model 102.” Ex. 1007, ¶31. Nevertheless, the existence of paths 108 and 110 does not affect my analysis here. The claims of the ’985 Patent only requires “using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm,” but does not preclude other information passage between the first and second machine learning algorithms.

229. During runtime, Chalom’s RFC 306 (yellow, below) would simply be replaced with the newly trained student model (blue, above) with everything else unchanged.

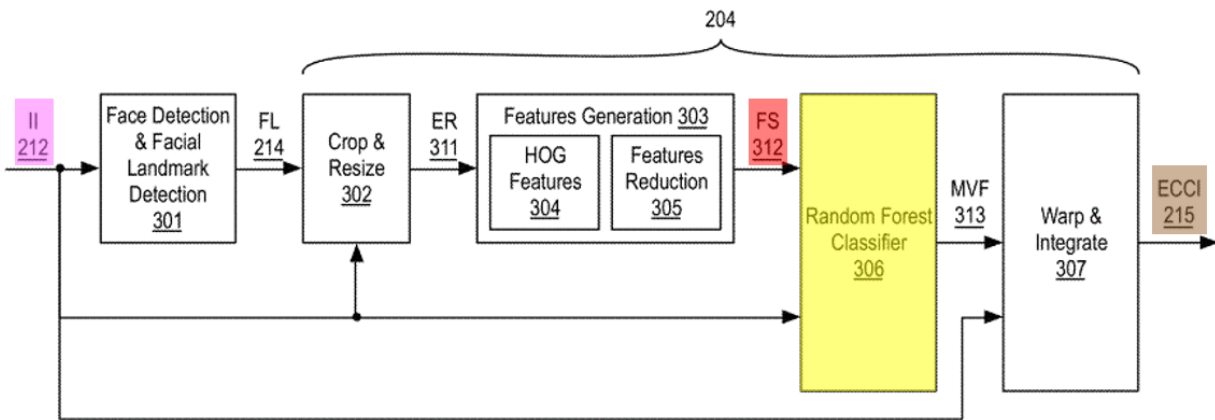
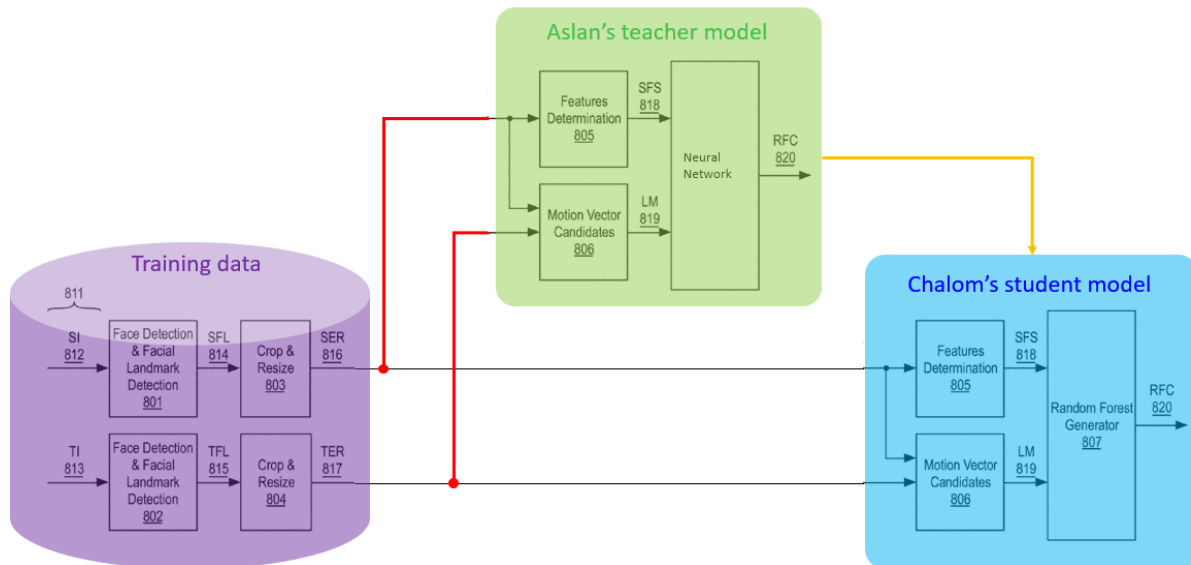


FIG. 3

Ex. 1005, Fig. 3.

230. In sum, it is my opinion that Aslan's student model is being incorporated into Chalom's training process according to known methods disclosed by Aslan and would have yielded predictable results.

231. Similarly, when using Chalom's RFC as a student, it is my opinion that a PHOSITA would still only have needed to follow the instructions provided in Aslan, such that before training Chalom's RFC (blue, below), use the same training data (purple) to first train a more accurate teacher model (green).



Ex. 1005, Fig. 8, modified based on Aslan to add a teacher model; *c.f.* Ex. 1007, Fig. 1. As I discussed above, a PHOSITA would have known that Aslan's teacher model does not necessarily have to include a features determination module 805 and a motion vector candidates module 806, or may share the same two modules with Chalom's student model. In this obvious combination, the same training data

(purple) and the output of the teacher model (orange) are provided to train Chalom's RFC (as student, blue). Thus, it is my opinion that, for this particular modification, Chalom's RFC is only trained differently, i.e., by learning from a teacher model, but no change is needed during runtime, i.e., no change to Chalom's Fig. 3 system is needed. Ex. 1005, Fig. 3.

232. Accordingly, in my opinion, a PHOSITA would have been motivated to incorporate Aslan's well-known teacher-student training architecture for training Chalom's RFC (using it as either the teacher or student model, as was explained above) to yield a working system.

2. Claim 1

233. In my opinion, claim 1 is unpatentable because it is rendered obvious by Chalom in view of Aslan. Claim 1 of the '985 Patent recites the following. I address each of these in my analysis below.

1[P] A method of generating reference data for adjusting a digital representation of a head region, the method comprising:

1[A] receiving training data comprising:

1[B] a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

1[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

1[D] using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and

1[E] using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

1[F] wherein the first reference data comprise first editing instructions for a range of possible configurations of the target feature and first selection instructions for selecting editing instructions for a particular input patch from the first editing instructions based on the configuration of the target feature of the input patch; and

1[G] wherein the second reference data comprise second editing instructions for a range of possible configurations of the target feature and second selection instructions for selecting editing instructions for a particular input patch from the second image editing instructions based on the configuration of the target feature of the input patch.

a. 1[P]

234. If the preamble is limiting, it is my opinion that Chalom discloses it.

235. The '985 Patent describes generating reference data with respect to Fig. 8, reproduced below, which illustrates a **training process** using “training data” 100 (purple).

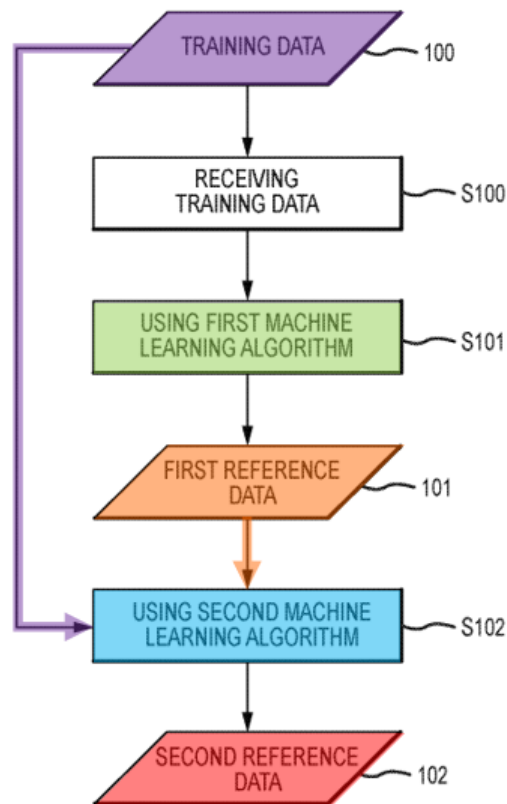


FIG.8

Ex. 1001, Fig. 8, 3:63-64 (“FIG. 8 is flow chart of a method of generating reference data.”). As shown, the method in Fig. 8 generates “first reference data 101” (orange) and “second reference data 102” (red). For reasons I will explain

below for Limitations 1[D-E] in Section XII.C.2.e, Chalom discloses generating the “first/second reference data” and therefore discloses “[a] method of generating reference data” required by this preamble.

236. Claim 1 also requires that the generated reference data is “for adjusting a digital representation of a head region.” According to the ’985 Patent, “adjusting a digital representation of a head region” could be “**correct[ing] a perceived gaze direction of eyes.**” Ex. 1001, 1:9-12. As shown in Fig. 3, reproduced below, reference data 35 (red) is used to perform gaze correction (step 5, green).

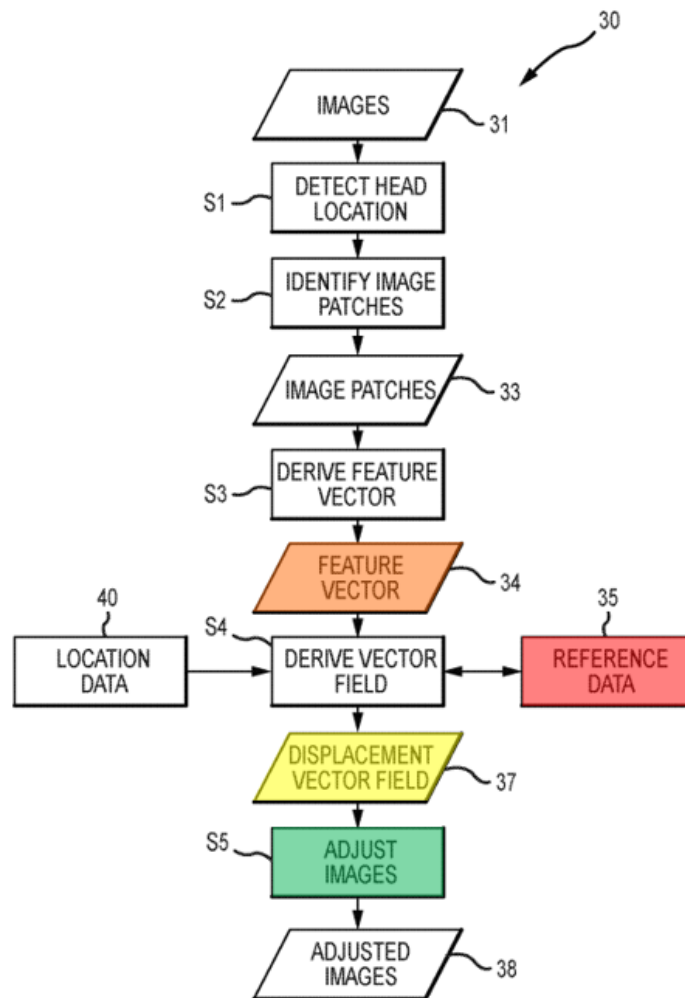


FIG.3

Ex. 1001, Fig. 3. Specifically, “[i]n step S4, at least one displacement vector field 37 [yellow] representing a transformation of an image patch is derived by using the feature vector 34 [orange]... to look up the reference data 35 [red].” *Id.*, 11:38-41. “The reference data 35 may comprise reference displacement vector fields that represent transformations of the image patch.” *Id.*, 7:48-50. The derived displacement vector field 37 (yellow) is then used to perform the gaze correction,

shown as step S5 (green).

237. 1[P] recites “[a] method of generating reference data **for adjusting** a digital representation of a head region” and the remaining elements of claim 1 recite generating editing instructions for adjusting a digital representation of a head region **using training data**. Thus, in my opinion, a PHOSITA would have understood that claim 1 is for a **training process** for gaze correction (e.g., as depicted in Fig. 8 of the ’985 Patent), which occurs before the runtime gaze-adjustment process (e.g., as depicted in Fig. 3 of the ’985 Patent). *See, e.g.*, Ex. 1001, 7:55-56.

238. Like the ’985 Patent, Chalom discloses “a **training component** (e.g., performed offline) and a **real-time component** (e.g., performed during implementation or run time) that provides **eye contact correction** [i.e., adjusting a digital representation of a head region].” Ex. 1005, 3:21-24.

239. During runtime, as shown in Fig. 3 of Chalom, which I reproduce below, RFC 306 (yellow) “receive[s] feature sets 312 [red]” and “each feature set of feature sets 312... traverses branches of the forest based on decisions at each branch until a leaf is reached such that the leaf corresponds to or provides a motion vector field [e.g., motion vector field 313 (green)] for the feature set.” Ex. 1005, 9:17-22.

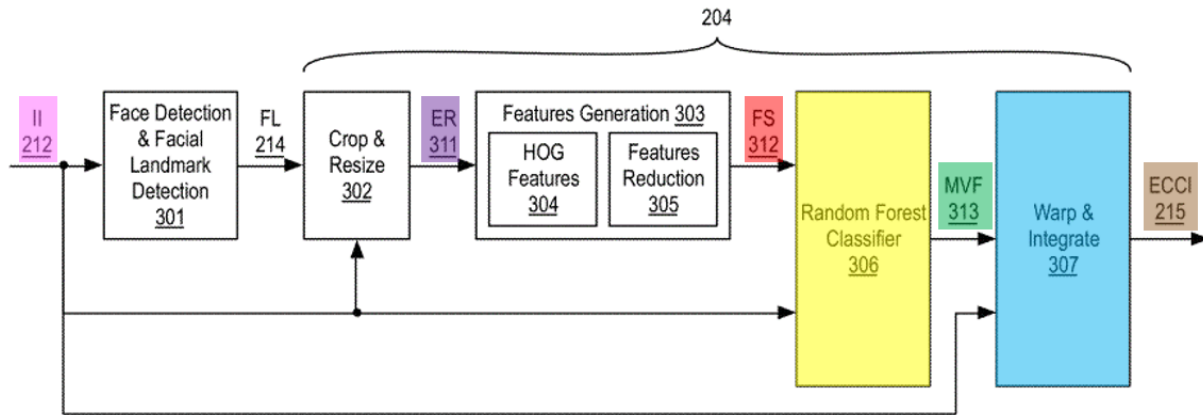


FIG. 3

Id., Fig. 3. The input image 212 (pink) may then be warped by the warp and integrate module 307 (blue) “based on motion vector fields 313 [green].” Ex. 1005, 9:26-30.

240. Thus, as I discussed in Section V.D.2, Chalom’s RFC has multiple decision trees and each tree has multiple leaves, each leaf representing a motion vector field distribution (i.e., motion vector field likelihood map) over the motion vectors for the training samples (i.e., input eye regions from training data) that reached that leaf during training. During runtime, an input eye region is provided to each decision tree and a motion vector field distribution is obtained from each tree. Finally, the motion vector field distributions from all the decision trees are aggregated and the final optimal motion vector field for the input eye region is selected to be the most probable one based on the obtained aggregated motion vector field distribution (i.e., the one that maximizes the aggregate motion vector

field distribution, e.g., likelihood of occurrence; alternatively, it is the one that minimizes the displacement error aggregate distribution, e.g., SAD map). *See also* Ex. 1028, p. 4.

241. Chalom further describes the **training process** for its RFC 306 in Figure 8, which I reproduce below.

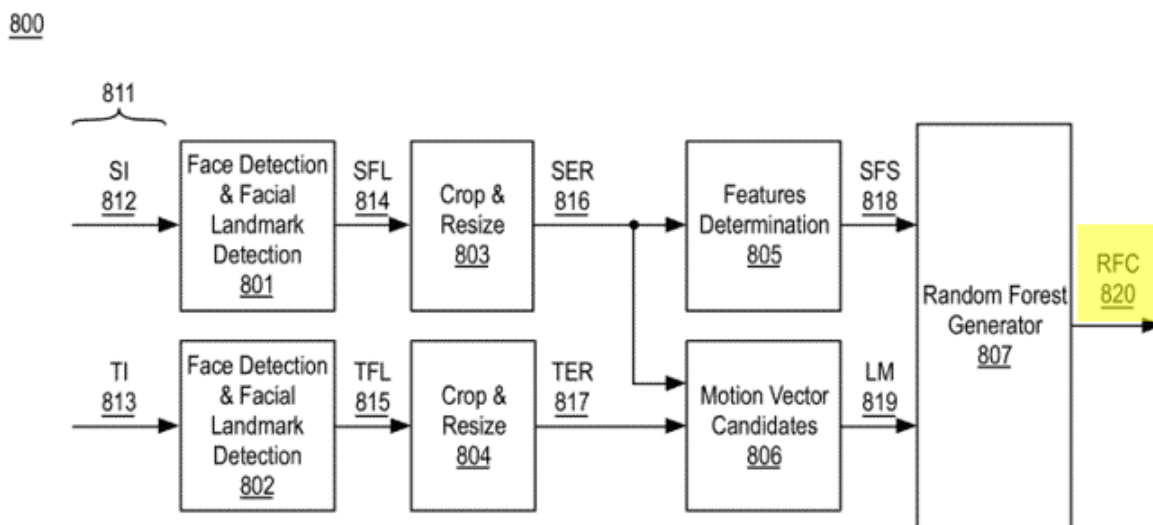


FIG. 8

Ex. 1005, Fig. 8. Chalom discloses that “[t]raining may be performed such that **each leaf of a tree in random forest classifier 820 [yellow] represents a [motion vector] likelihood map (e.g., a SAD map) for every pixel in a source eye region**” and that “source features sets 818 and likelihood maps 819 [both of which are determined from the training data SER 816, extracted from SI 812, and/or TER 817, extracted from TI 813, in Fig. 8] may be provided to random forest generator

module 807, which may train and determine random forest classifier 820 based on source feature sets 818 [extracted from the training data SER 816] and likelihood maps 819 [derived using the training data SER 816 and TER 817].” *Id.*, 13:41-55, 9:21-22. Furthermore, Chalom discloses that, during the training process, the **“random forest classifier 820 may be trained to determine optimal motion vector fields [for the input training data] from [the training data’s] source feature sets 818.”** *Id.*, 13:49-52, Fig. 8. Chalom also discloses that such **training process** results in a trained RFC that generalizes to unseen data samples and **“determines an optimal motion vector field** based on the feature set [of input data] during the **implementation phase.”** *Id.*, 4:58-60. Additionally, it is my opinion that a PHOSITA would have understood that in order for the RFC to be able to generate an optimal motion vector field during the implementation phase, optimal motion vector fields are also generated for the training data during the training process. *See infra* Section XII.C.2.e.

242. Therefore, in my opinion, Chalom discloses “[a] method of generating reference data [e.g., motion vector fields generated during training, including the recited “first reference data” in 1[D] and the recited “second reference data” of 1[E]] for adjusting a digital representation of a head region [e.g., gaze correction].”

b. 1[A]

243. 1[A] requires “receiving training data,” which, in my opinion, is disclosed by Chalom.

244. For example, Chalom discloses that “during a training phase of the principal component analysis and/or the pretrained classifier, training may be performed based on **a training set of images** having known gaze angle difference therebetween.” Ex. 1005, 4:19-22. “During training, the training set images may be cropped to eye regions and the eye regions may be aligned to provide **pairs of training eye region images.**” *Id.*, 4:25-28; Fig. 8 (receiving source images 812 and target images 813 (pink), cropped and resized to generate source eye regions 816 and target eye regions 817 (purple)).

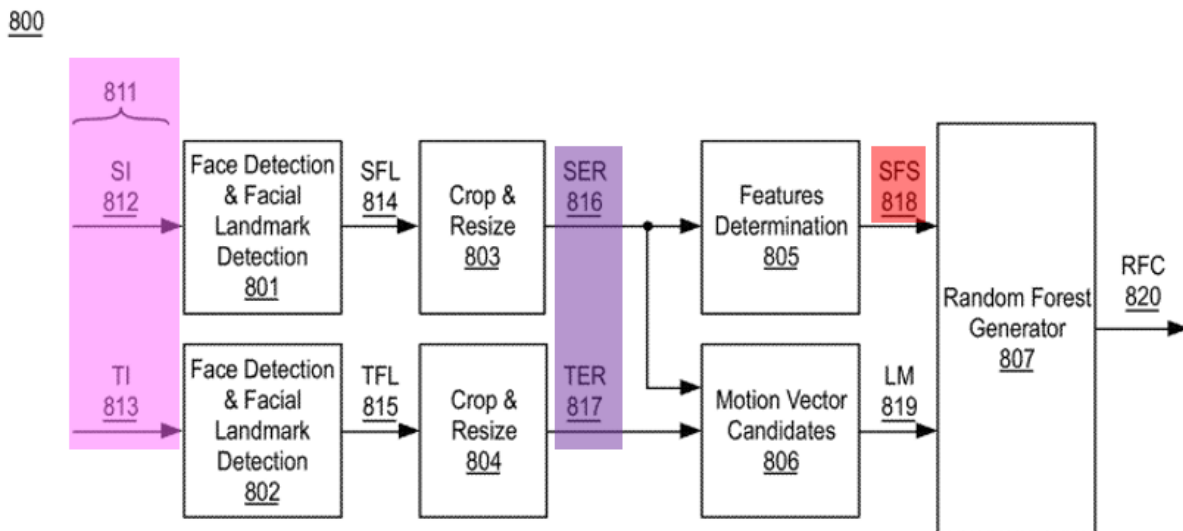


FIG. 8

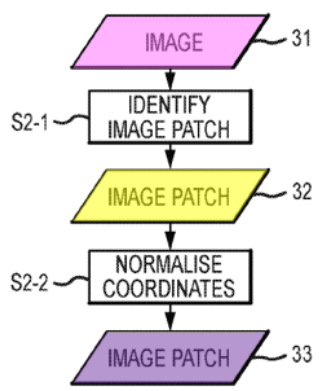
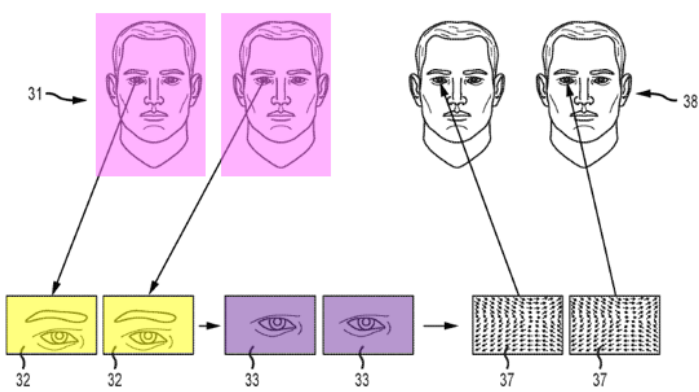
Id., Fig. 8.

245. Therefore, in my opinion, Chalom discloses “receiving training data [e.g., pairs of training eye region images, including source eye regions 816 and target eye regions 817].”

c. 1[B]

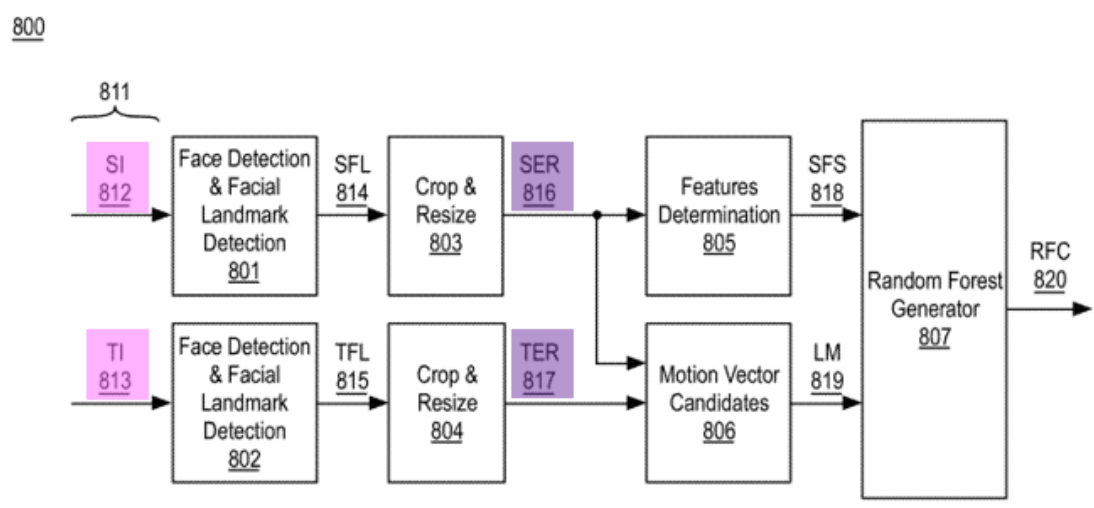
246. According to the '985 Patent, “[t]he training input to the machine learning technique is two sets of images (or image patches).” Ex. 1001, 7:65-66. **“The first set are input images**, being images of each individual **with an incorrect gaze** where the error is known a priori.” *Id.*, 8:6-8. **“The second set are output images**, being images of each individual **with correct gaze** for a predetermined observer location relative to a display location in which the image is to be displayed.” *Id.*, 8:15-18. In addition, a “target feature comprises **an eye region** comprising at least part of an eye and the adjustment of the digital representation (e.g. image) of the head region comprises **adjusting a gaze direction**.” *Id.*, 13:58-61. Therefore, 1[B] concerns **a set of input image patches**, each having an eye region **before gaze correction**.

247. According to the '985 Patent, image patches 32 (yellow) are first obtained based on facial landmarks in step S2-1 (yellow), and then normalized to obtain image patch 33 (purple) in step S2-2, which is used for training machine learning algorithms. *Id.*, 7:8-17.



Id., Figs. 4 and 5.

248. Like the '985 Patent, Chalom's training data includes two sets of image patches. As shown below, Chalom's training data "include pairs of **source images 812** and target images 813 [pink]." Ex. 1005, 11:24-26.



Id., Fig. 8. Chalom's source images 812 and target images 813 (pink) are analyzed by "face detection and facial landmark detection modules 801, 802" and then

cropped and resized (803 and 804) to form source eye regions 816 and target eye regions 817 (purple), both of which are then used for training. *Id.*, 11:13-14.

Since face and landmark detection and cropping and resizing are performed on each of the source images 912 and target images 913, they each have the same target feature, e.g., an eye region, of a digital representation of a head region. *Id.*, 12:1-22.

249. Therefore, in my opinion, Chalom discloses “a set of input patches [e.g., source eye regions 816], each input patch comprising a target feature [e.g., eye region] of a digital representation [e.g., image] of a head region prior to adjustment [e.g., before gaze correction] of the digital representation of the head region, wherein the target feature is the same for each input patch.”

d. 1[C]

250. 1[C] recites “a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region,” i.e., a set of output patches that are in one-to-one correspondence with the input patches discussed above. In my opinion, Chalom teaches this.

251. As I explained in Section XII.C.2.c above, Chalom’s training data includes a set of output patches, e.g., target eye regions 817 (purple) obtained by

cropping and resizing the target images 813 (pink) after face & landmark detection 802.

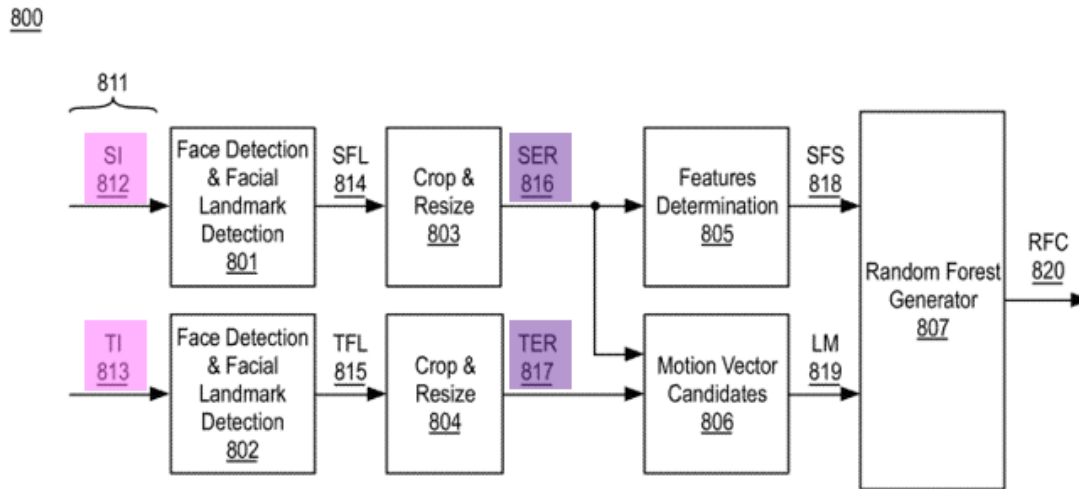


FIG. 8

Ex. 1005, Fig. 8.

252. In the '985 Patent, the output patches and the input patches are in one-to-one correspondence because “in the case where an image of the head region is **to be adjusted** to correct a gaze direction, each output patch comprises **an eye region that has been adjusted** so that the gaze appears to be in the desired direction.” Ex. 1001, 14:40-43. In particular, the gaze difference between each input and output patches is known. *Id.*, 8:6-8 (“The first set are input images, being images of each individual with **an incorrect gaze where the error is known a priori.**”).

253. Chalom discloses the same. As shown in Fig. 8 above, source image 812 and target image 813 (pink) correspond to each other such that source image 812 includes “a user looking in a downward direction” (i.e., gaze to be adjusted) and the corresponding target image 813 includes the same user “looking at the camera” (i.e., gaze has been adjusted in the desired direction). Ex. 1005, 11:36-39, 12:10-22, Fig. 11.

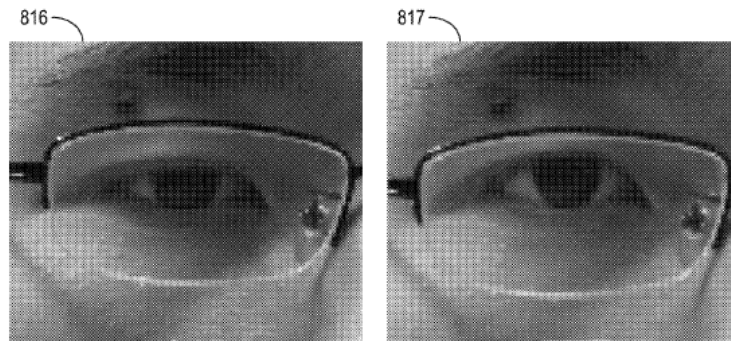


FIG. 11

Id., Fig. 11.

254. Chalom further discloses that “source images 812 and target images 813 may include any number of pairs of training images in analogy to those illustrated in FIG. 9.” Ex. 1005, 11:43-45.

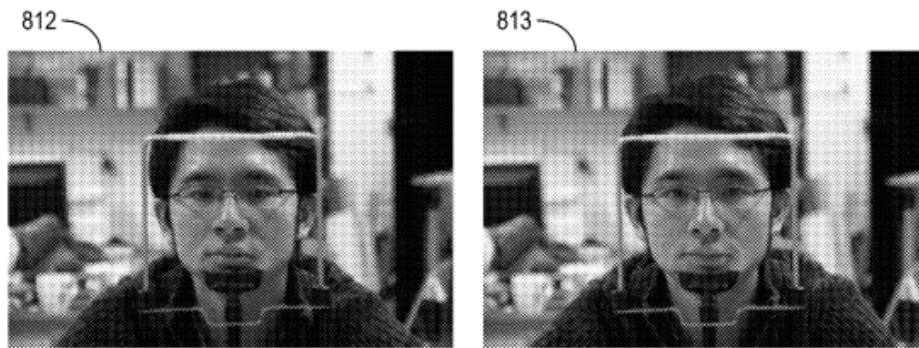


FIG. 9

Id., Fig. 9. Chalom also discloses that “in addition to cropping and resizing, system 800 [Fig. 8] may align source eye regions 816 and target eye regions 817.” *Id.*, 12:18-20. Thus, it is my opinion that after going through the same face and landmark detection and cropping and resizing processes, the obtained source eye regions 816 and target eye regions 817 are also in one-to-one correspondence.

255. Therefore, in my opinion, Chalom discloses “a set of output patches [e.g., target eye regions 817] in one-to-one correspondence with the input patches, each output patch comprising the target feature [e.g., eye region] of the digital representation of the head region after adjustment [e.g., after gaze correction, e.g., “with correct gaze”] of the digital representation of the head region.”

e. 1[D-E]

256. 1[D] and 1[E] collectively require “using a **first machine learning algorithm** to generate **first reference data using the training data**, the first reference data comprising **editing instructions** for adjusting the digital

representation of the head region for a range of possible digital representations of the head region,” as well as “using a **second machine learning algorithm** to **generate second reference data** using the **same training data as the first machine learning algorithm** and **the first reference data generated by the first machine learning algorithm**, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,” which, in my opinion, are disclosed by the Chalom-Aslan combination.

257. First, it is my opinion that Chalom discloses “using a [] **machine learning algorithm** to generate [] reference data using the training data, the [] reference data comprising **editing instructions for adjusting** the digital representation of the head region” required by 1[D-E].

258. As shown in Fig. 8 of Chalom below, training data such as source eye regions 816 and target eye regions 817 are used by system 800 “to generate a random forest classifier 820.” Ex. 1005, 11:29-30.

800

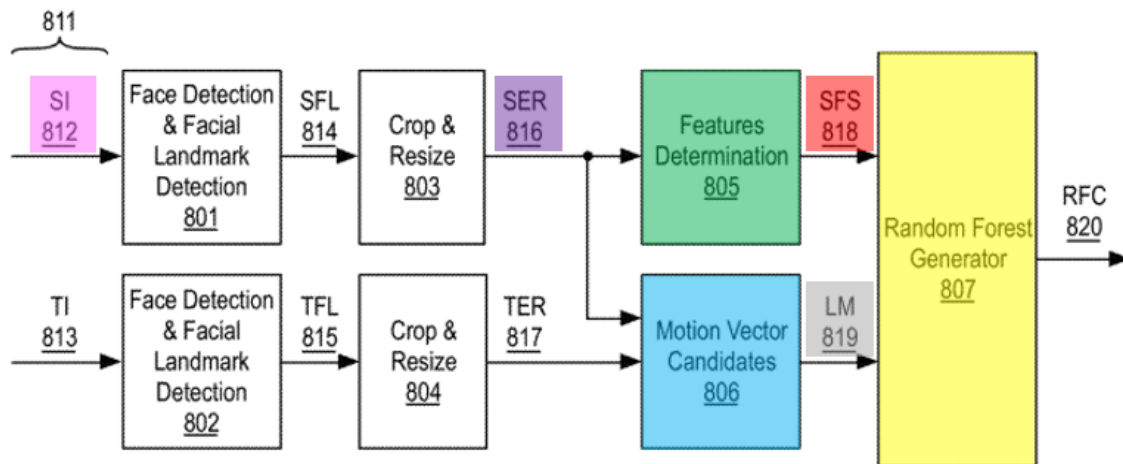


FIG. 8

Id., Fig. 8. Specifically, “features determination module 805 [green] may receive source eye regions 816 [purple]... [and] determine source feature sets (SFS) 818 [red].” *Id.*, 13:8-11. Motion vector candidates module 806 (blue) receives both source eye regions 816 and target eye regions 817 to generate candidate motion vectors and likelihood maps 819 (gray). *Id.*, 12:23-27. The source feature sets 818 and likelihood maps 819 are then provided to random forest generator module 807 (yellow) to “train and determine random forest classifier 820.” *Id.*, 13:45-49.

259. Chalom’s trained RFC 820 comprises “tree[s]” each having multiple “branches” and “lea[ves].” Ex. 1005, 9:9-22, 13:52-55. “[E]ach leaf of the tree of the random forest classifier may **represent a likelihood map** (e.g., a SAD map) for every pixel in the eye region” and “corresponds to or provides a **motion vector field** for the feature set.” *Id.*, 4:62-64, 9:21-22, 13:49-55. The “motion vector

field may be used to **warp the eye region** to generate the desired output image having corrected eye contact.” *Id.*, 3:36-38. Thus, Chalom’s training results in each leaf of the trees in Chalom’s RFC 820 providing a motion likelihood map for every pixel in source eye region and a corresponding motion vector field.

260. Therefore, it is my opinion that Chalom discloses “using a [] **machine learning algorithm** [e.g., Chalom’s RFC] to generate [] reference data using the training data [e.g., Chalom’s source eye regions 816 and target eye regions 817], the [] reference data comprising **editing instructions** [e.g., likelihood maps and motion vector fields as “leaves” of the RFC] for adjusting the digital representation of the head region [e.g., gaze correction, e.g., warping the eye region].”

261. Second, it is also my opinion that Chalom discloses 1[D]’s further recitation of “adjusting the digital representation of the head region for **a range of possible digital representations of the head region.**”

262. According to the ’985 Patent, “a range of possible digital representations [] of the head region” could be “digital representations representing different states of the head, such as different positions and/or orientations.” Ex. 1001, 14:50-54. Chalom discloses the same by teaching that its “[eye] contact correction module [] may be capable of implementing eye contact correction for **multiple orientations or relative positions between camera 104 and display 102.**” Ex. 1005, 10:12-14. To allow for such various “relative position[s] between

camera 104 and display 102,” “camera 104 may be moved from a central position of display 102 to a position off center but still within an edge of display 102,” for example. *Id.*, 5:40-50. Thus, when moving camera 104 relative to display 102, with the person’s head region being captured by camera 104, at least the “orientations” of the person’s head would change. *Id.*, 10:10-14.

263. Thus, in my opinion, Chalom further discloses “adjusting the digital representation of the head region for a range of possible digital representations of the head region [e.g., different head orientations].”

264. Third, as to the use of a second MLA, it is my opinion that Chalom-Aslan discloses this as well. As I explained earlier, it would have been obvious for a PHOSITA to modify Chalom’s system to use Aslan’s teacher-student architecture, which includes a teacher MLA and a student MLA, such that Chalom’s training processes then involve use of two MLAs.

265. In my opinion, although the ’985 Patent does not use the exact term of “teacher-student,” it is directed to such an implementation.

266. For example, as shown in Fig. 8 of the ’985 Patent, the training process involves training two machine learning algorithms, i.e., the first machine learning algorithm S101 (**MLA1**, green) and the second machine learning algorithm S102 (**MLA2**, blue). Ex. 1001, 14:45-67.

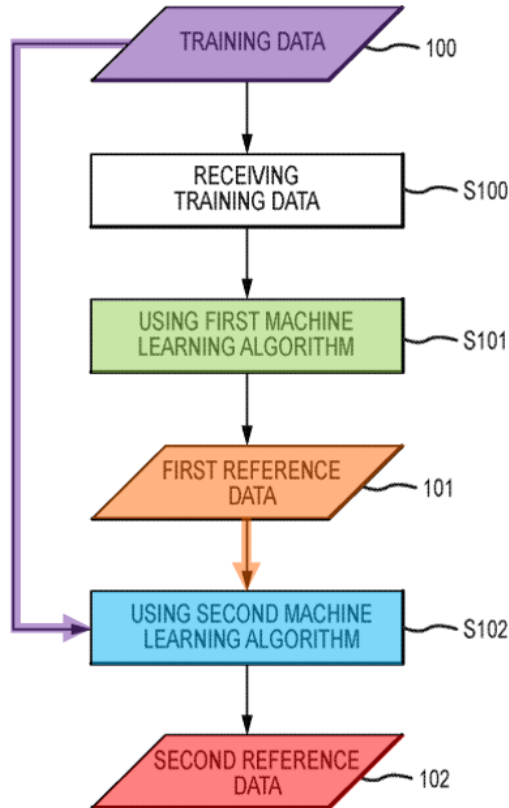


FIG.8

Ex. 1001, Fig. 8. Specifically, MLA1 may be “a **neural network** (known to provide relatively high accuracy, but at the expense of relatively high computational demands)” and MLA2 may be “a **regression forest** (known to provide higher computational efficiency, but at the expense of reduced accuracy).” *Id.*, 16:24-30. As illustrated in Fig. 8 above, “the combination of the two different machine learning algorithms” is realized by training MLA2 to generated second reference data 102 using “the same training data 100” as MLA1 and “the first

reference data 101” output by MLA1. *Id.*, 14:55-62. In this way, MLA2 essentially “learns” MLA1, by using the first reference data as ground truth, so as to “provide [second] reference data that can be used in an adjustment method with high efficiency while still achieving high adjustment accuracy.” *Id.*, 16:30-34.

267. The '985 Patent's teacher-student architecture is further illustrated in its Fig. 9 below.

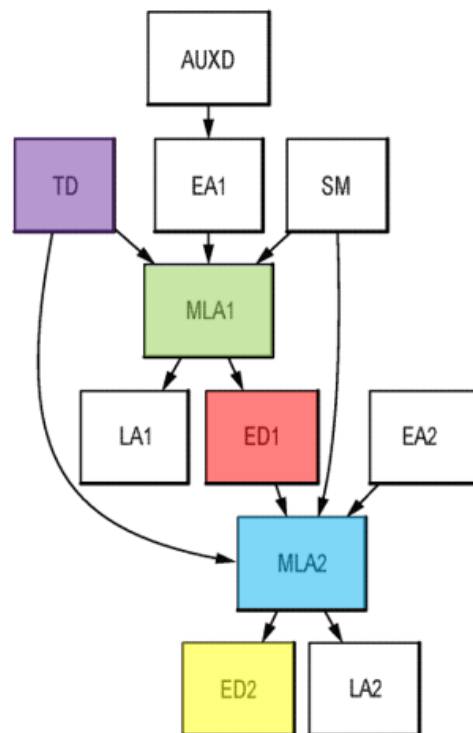


FIG.9

Ex. 1001, Fig. 9. Here, MLA1 (green) receives training data (TD, purple) and “outputs... first editing instructions ED1 [red].” *Id.*, 17:5-6. Training data TD

includes both “input patches” and “output patches,” where the “input patches” modified by ED1 are compared to the “output patches,” the differences therebetween are minimized during training of MLA1. *Id.*, 16:48-54. MLA2 then “learns” from MLA1 by receiving the “input patches” of the training data and ED1. *Id.*, 17:14-17. In this way, “MLA2 thus gets the editing instructions to match or to approximate, and does not have to infer these from the matched input images as MLA1.” *Id.*, 17:14-17. In other words, when training MLA2, MLA2 is trained to output ED2 such that ED2 matches or approximates ED1 as much as possible and does not need to infer those by comparing and minimizing the difference between the ED2-modified “input patches” and the ground-truth “output patches” as has been done when training MLA1. *See also, e.g.*, Ex. 1011, p. 6 (“MLA2 learns the editing information directly. By making the desired output explicit, MLA2 has a better chance at fitting the output.”). In this way, the more-efficient-but-less accurate MLA2 is trained to learn from the less-efficient-but-more-accurate MLA1, and the resulting MLA2 provides “high efficiency while still achieving high adjustment accuracy.” Ex. 1001, 16:24-34. Thus, MLA1 is only used for training MLA2, and during runtime, only the trained MLA2 is used for gaze correction. *Id.*, Fig. 12 (“second editing instructions,” i.e., “ED2”, 18:39-42 (“The input data ID is input to the second selection instructions LA2 to select editing instructions appropriate to the input data ID from the second editing

instructions ED2.”).

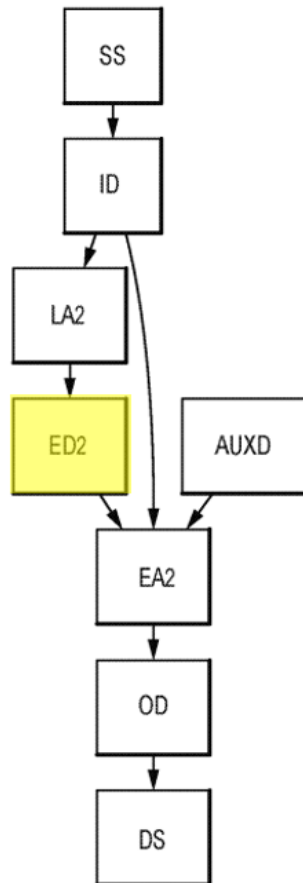


FIG. 12

Id., Fig. 12. However, I would like to note that in a teacher-student architecture, training of the MLA2 may still utilize the ground-truth output patches, e.g., to further improve accuracy. This is taught by Aslan. *See* Ex. 1007, ¶¶35-36, Eq. 2.

268. Like the '985 Patent, as shown in Fig. 1 of Aslan below, during training, both the teacher model 100 (green) and the student model 102 (blue) in Aslan “receiv[e], or hav[e] access to, the [same] training data 104.” Ex. 1007, ¶23.

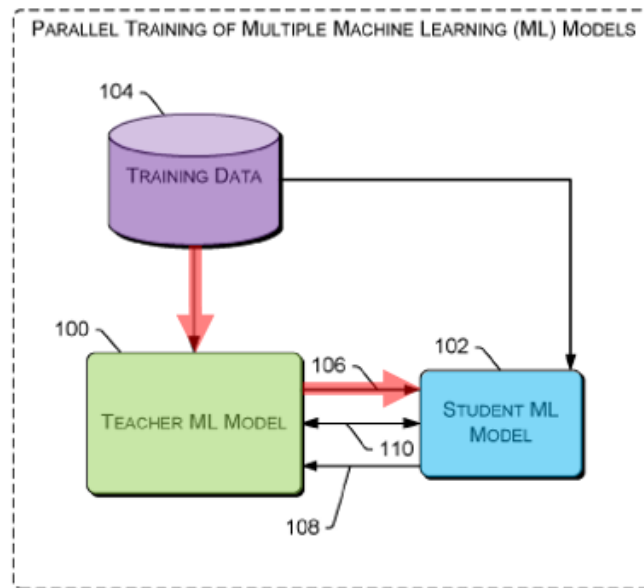


Fig. 1

Ex. 1007, Fig. 1. In the combined system, the training data would have been the source eye regions and target eye regions from Chalom, and this training data would have covered a range of possible digital representations of the head region based on the different camera angles utilized (as I explained above).

269. According to Aslan, the teacher model and the student model “can be trained in parallel so that each model learns a task” and “the ‘task’ can comprise a task to infer an expected output based at least in part on an unknown input,” such

as “a binary classification task,” “a multi-label classification task,” or “to infer a set of probabilities based on unknown input data.” Ex. 1007, ¶25. The task learned by teacher and student models “can be the same task.” *Id.*

270. The teacher model “can comprise a large, complex ensemble of machine learning models that is often too large and/or slow to be used at run-time in particular scenarios”; the student model “can comprise a much smaller machine learning model (e.g., a neural net with 1000 times fewer parameters than the first model 100) that has the size and/or speed that is advantageous at run-time in particular scenarios.” Ex. 1007, ¶44. By “joint training” the teacher and student models, the student model “can be trained to mimic the much larger [teacher model]... through **learning how to approximate the function** learned by” the teacher model, such that the student model “can be a compressed formed of” the teacher model and “can be readily deployed on computer devices with limited resources (e.g., mobile devices, wearables, etc.).” *Id.* As shown in Figure 6 below, such “learning... to approximate” is realized by “passing information” (i.e., “transfer of knowledge”) between the teacher model and the student model (yellow). *Id.*, ¶6.

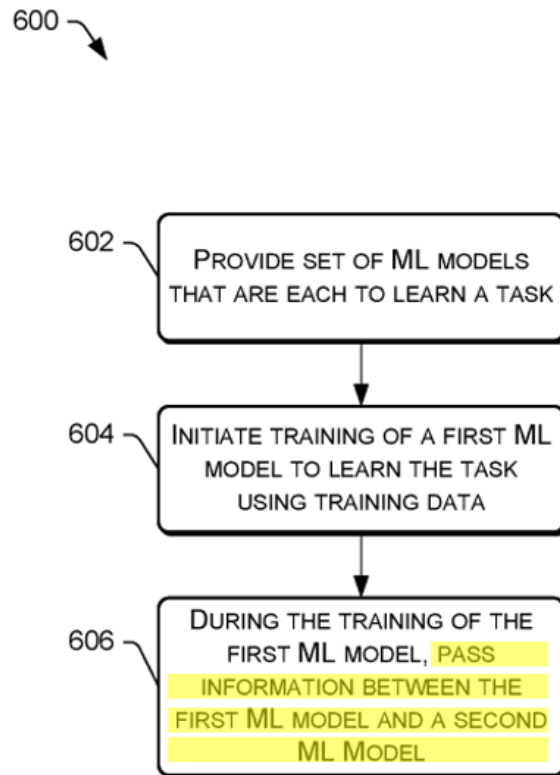


Fig. 6

Id., Fig. 6. “In this manner, the second machine learning model can **access information about the outputs of the first machine learning model** based on the first model’s processing of the training data as input prior to the first model completing its training.” *Id.*, ¶6. This is “depicted visually in FIG. 1 by the path 106 [red] that goes from the training data 104 to the first model 100, and from the first model 100 to the second model 102.” *Id.*, ¶26.

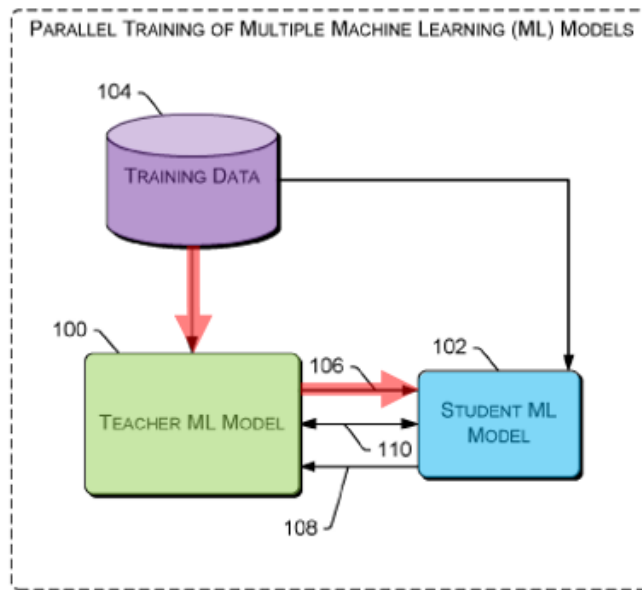


Fig. 1

Id., Fig. 1. Here, “the first model 100 can learn from the training data 104, and **the training of the second model 102 can be influenced by what the first model 100 is learning from the training data 104** while the first model 100 is being trained, and/or before the first model 100 completes its training. In this sense, **the second (student) model 102 can be considered to be learning from the first (teacher) model 100** as the first model 100 learns.” *Id.*, ¶26.

271. As I explained earlier, it is my opinion that a PHOSITA would have been motivated to incorporate Aslan’s teacher-student architecture to improve Chalom’s RFC in at least two different ways. *Supra* Section XII.C.1. Regardless of how Aslan’s teacher-student architecture is incorporated into Chalom, it is my

opinion that the Chalom-Aslan system includes a teacher model and a student model. According to Aslan, both the teacher and student models may be trained to perform the same task based on the same training data (the source and target eye regions from Chalom). Ex. 1007, ¶25.

272. In my opinion, a PHOSITA would have understood that the output of the teacher (which is provided to the student) and the output of the student are of the same type. *See* Ex. 1007, ¶31 (matching teacher’s output). In the Chalom-Aslan system, the “task” is generating motion vector fields for gaze correction, and the teacher’s output, e.g., first editing instructions including motion vector fields, is used to train the student that also generates motion vector fields, i.e., second editing instructions.

273. To the extent that Patent Owner argues that Aslan discloses parallel training of teacher and student models, I note that Aslan’s parallel training “initiate[s] training” of the teacher model, during which “information can be passed” to the student model. Ex. 1007, ¶6. To the extent that Patent Owner argues that training of the teacher model must complete before training of the student model starts, Aslan also discusses sequential training of teacher and student models, such that it was known to use sequential or parallel training. *Id.*, ¶5. Additionally, I note that the claims of the ’985 Patent do not require sequential learning and thus they are taught by parallel or sequential learning.

274. Therefore, in my opinion, the Chalom-Aslan combination discloses “using a first machine learning algorithm [e.g., Chalom-Aslan’s teacher model] to generate first reference data using the training data, the first reference data comprising editing instructions [e.g., motion vector fields generated by the teacher model] for adjusting the digital representation of the head region [e.g., gaze correction] for a range of possible digital representations of the head region [e.g., different head orientations],” as well as “using a second machine learning algorithm [e.g., Chalom-Aslan’s student model] to generate second reference data using the same training data as the first machine learning algorithm [e.g., training data for the teacher model] and the first reference data generated by the first machine learning algorithm [e.g., motion vector fields generated by the teacher model], the second reference data comprising editing instructions [e.g., motion vector fields generated by the student model] for adjusting the digital representation of the head region for a range of possible digital representations of the head region.”

f. 1[F-G]

275. The claim requires that “the first reference data comprise first editing instructions for a range of possible configurations of the target feature and first selection instructions for selecting editing instructions for a particular input patch from the first editing instructions based on the configuration of the target feature of

the input patch,” as well as “the second reference data comprise second editing instructions for a range of possible configurations of the target feature and second selection instructions for selecting editing instructions for a particular input patch from the second image editing instructions based on the configuration of the target feature of the input patch,” which, in my opinion, are taught by the Chalom-Aslan combination.

276. First, as I explained in Section XII.C.2.e above, Chalom discloses generating reference data that comprises editing instructions, e.g., motion vector fields, for gaze correction, during training.

277. It is also my opinion that Chalom’s motion vector fields generated during training are “for a range of possible configurations of the target feature.” According to the ’985 Patent, “a range of possible configurations of the target feature” could be “different gaze directions.” Ex. 1001, 15:1-5. In my opinion, Chalom discloses the same. For example, Chalom discloses “Receiv[ing] Pairs of Training Eye Images having Different Gaze Angles.” Ex. 1005, Fig. 14 (operation 1401). Chalom discloses that “random forest classifier 306 may apply **different random forest models** responsive to or selectively based on camera 104 and display 102 **having a particular relative position therebetween**,” Ex. 1005, 10:35-39, which thereby results in having training data that expressly, or at least inherently includes different gaze directions.

278. In my opinion, a PHOSITA would have understood that when the relative position between camera 104 and display 102 changes, with a person continuing to look at display 102, the gaze direction captured by camera 104 and the magnitude of gaze correction both change. Thus, in my opinion, for Chalom's RFC 306 to be able to correct gaze for different input gaze directions, a PHOSITA would have understood that the training data necessarily includes eye regions with different gaze directions. Accordingly, it is my opinion that the editing instructions generated during training using such training data would be "for a range of possible configurations of the target feature," e.g., different gaze directions.

279. As I also explained for in Section XII.C.2.e above, after incorporating Aslan's teacher-student architecture in Chalom, both the teacher model and the student model each generate respective motion vector fields for gaze correction, i.e., first editing instructions and second editing instructions respectively, and the teacher's motion vector fields are passed to train the student. Thus, it is my opinion that a PHOSITA would have understood that, in Chalom-Aslan, since the training data for both the teacher and student models are the same, the first editing instructions and second editing instructions generated from them respectively are both "for a range of possible configurations of the target feature," e.g., different gaze directions.

280. Second, in my opinion, Chalom in view of Aslan also discloses aspects related to first and second selection instructions and their intended uses for selecting respective editing instructions, as recited in 1[F-G].

281. While claim 1 is directed to a machine-learning training process, it nonetheless recites intended uses of the outputs of the training process for runtime purposes. For example, 1[F] recites the “first reference data” generated by the “second machine learning algorithm” **during training**, but also recites that “the first reference data comprise... first selection instructions **for selecting editing instructions** for a particular input patch from the first editing instructions based on the configuration of the target feature of the input patch,” which reflects how generated “first reference data” comprising “first selection instructions” is intended to be used **during runtime**. *See also* 1[G] (“second selection instructions **for selecting editing instructions...**”). To the extent that these intended-use recitations are limiting, it is my opinion that Chalom-Aslan discloses them.

282. In Chalom, “each leaf of the tree of the random forest classifier may represent a likelihood map (e.g., a SAD map) for every pixel in the eye region” and “corresponds to or provides a motion vector field.” Ex. 1005, 4:60-64, 9:21-22. Therefore, the training of Chalom’s RFC generates not only motion vector fields, but also the trained RFC 820 (yellow) as shown in Fig. 8 below (and RFC 306 in

Fig. 3), represented by a group of trees having branches and leaves with each leaf corresponding to a motion vector field. *Id.*, 9:9-22, 13:52-55.

800

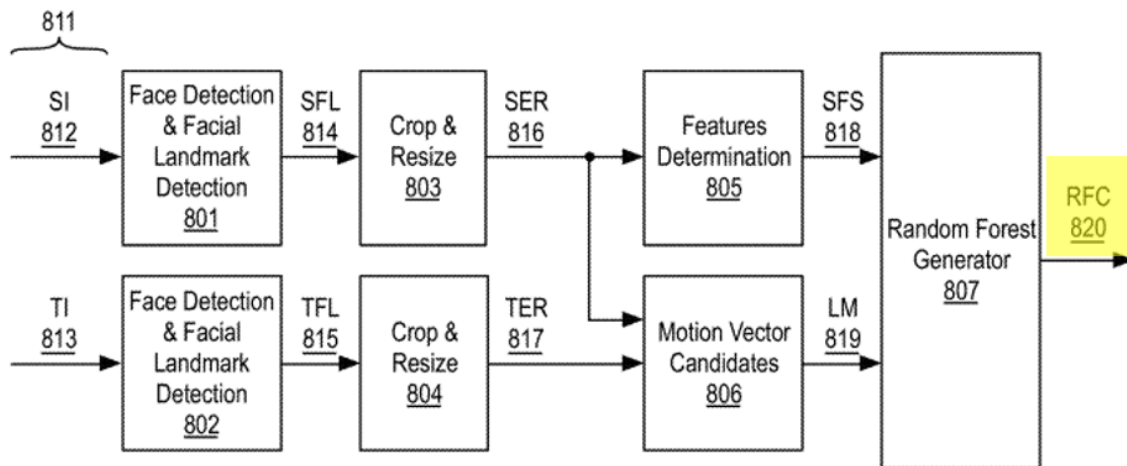


FIG. 8

Id., Fig. 8.

283. In addition, it is my opinion that, for a particular input source eye region, the corresponding motion vector field output by the trained RFC **during runtime** is selected from motion vector fields generated **during training**. This is because each leaf of Chalom’s RFC (generated during training) corresponds to a possible motion vector field, and the runtime implementation is the process of finding the optimal motion vector field among these leaves, i.e., possible motion vector fields. Thus, in my opinion, Chalom’s trained RFC provides “selection instructions for selecting editing instructions [e.g., optimal motion vector field] for

a particular input patch from editing instructions [e.g., motion vector fields generated during training].”

284. Moreover, in my opinion, Chalom’s selection of an optimal motion vector field is based on “the configuration of the target feature of the input patch” required by 1[F-G]. Specifically, Chalom discloses extracting histogram of oriented gradient (HOG) features from input images, which are then “reduced using principal component analysis (PCA) techniques” to “convert[] the histogram of oriented gradient features to feature sets 312.” Ex. 1005, 8:57-61. As shown in Figure 3 below, Chalom’s selection of optimal motion vector fields (MVF 313, green) is based on the “feature set 312 [red],” i.e., configuration of the target feature of the input patch.

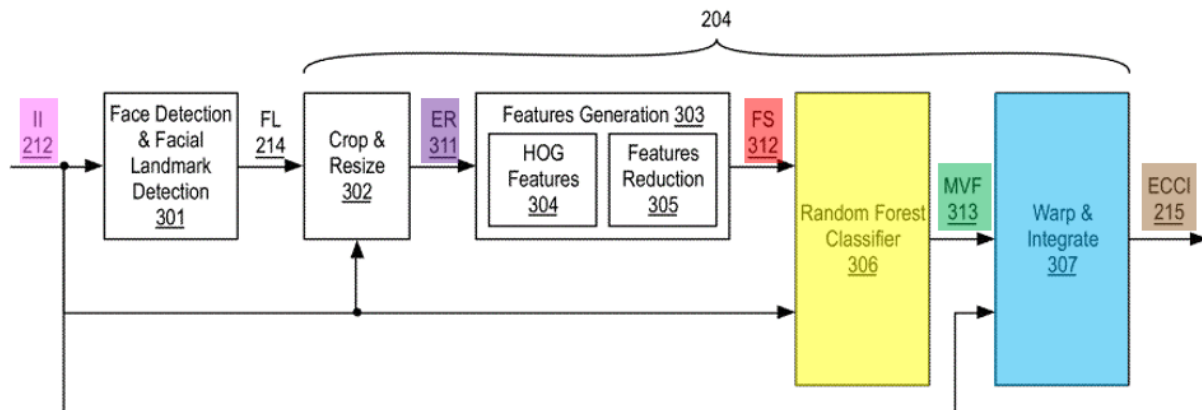


FIG. 3

Id., Fig. 3; claim 2.

285. It is my opinion that, after incorporating Aslan’s teacher-student training architecture, what has been discussed above with respect to Chalom equally applies to the teacher model and the student model—meaning, the teacher and student models each generate reference data that comprise “selection instructions for selecting editing instructions for a particular input patch from the [] editing instructions based on the configuration of the target feature of the input patch.” Although the teacher’s MLA may be different from the student’s MLA—for example, the teacher may be a neural network and the student may be an RFC—training of the teacher results in a trained teacher model and training of the student results in a trained student model. Even though the trained teacher and trained student may be represented by different sets of parameters (Sections V.D.1 and V.D.2), it is my opinion that both of them provide “selection instructions” for selecting the optimal motion vector fields from those generated during respective training.

286. Therefore, in my opinion, the Chalom-Aslan combination discloses that “the first reference data comprise first editing instructions [e.g., teacher’s motion vector fields generated during training] for a range of possible configurations of the target feature [e.g., different gaze directions] and first selection instructions [e.g., trained teacher model] for selecting editing instructions [e.g., optimal motion vector field] for a particular input patch from the first editing

instructions [e.g., teacher's motion vector fields generated during training] based on the configuration of the target feature of the input patch [e.g., Chalom's feature set 312],” and that “the second reference data comprise second editing instructions [e.g., student's motion vector fields generated during training] for a range of possible configurations of the target feature [e.g., different gaze directions] and second selection instructions [e.g., trained student model] for selecting editing instructions [e.g., optimal motion vector field] for a particular input patch from the second image editing instructions [e.g., student's motion vector fields generated during training] based on the configuration of the target feature of the input patch [e.g., Chalom's feature set 312].”

3. Claim 2

287. In my opinion, claim 2 is unpatentable because it is rendered obvious by Chalom-Aslan. Claim 2 of the '985 Patent recites the following. I address each of these in my analysis below.

2[P] The method of claim 1, wherein

2[A] the configuration of the target feature of each input patch is represented by a feature vector derived from plural local descriptors of the input patch, and;

2[B] the first and second selection instructions define how the feature vector is used to select editing instructions for the input patch.

288. First, it is my opinion that Chalom discloses that “the configuration of the target feature of each input patch is represented by a feature vector derived from plural local descriptors of the input patch” required by 2[A].

289. According to the '985 Patent, “local descriptors may include HOG features (Histogram of Oriented Gradients features).” Ex. 1001, 10:4-5. In fact, the '985 Patent acknowledges that “local descriptors” were well-known by incorporating a prior-art paper published in 2005. Ex. 1001, 10:5-9.

290. As I mentioned in Section XII.A.2, the '985 Patent admits that “local descriptors” were not new. Regardless, like the '985 Patent, Chalom also discloses extracting HOG features from input images. Chalom’s Fig. 3 discloses “a features generation module 303 that may include a histogram of oriented gradients (HOG) features module 304 [green],” “which may generate or determine histogram of oriented gradient features for each eye region of eye regions 311.” Ex. 1005, 7:27-29, 8:30-34.

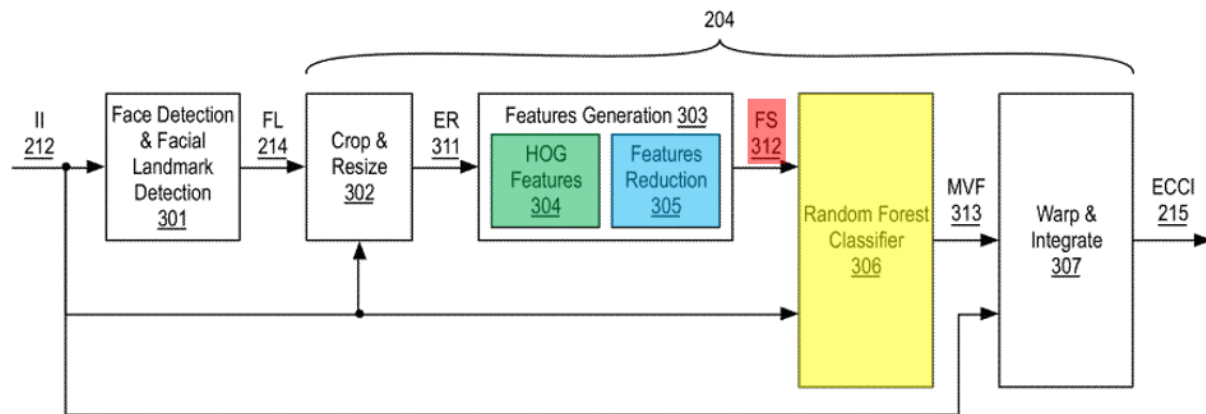


FIG. 3

Id., Fig. 3.

291. In my opinion, Chalom further discloses a feature vector generated from these HOG features. As shown in Fig. 3 above, “[t]he histogram of oriented gradient features generated by features generation module 303 may be reduced by features reduction module 305 [blue].” Ex. 1005, 8:47-51. The HOG features “may be **reduced using principal component analysis (PCA)** techniques... [which] may include converting the histogram of oriented gradient features to **feature sets 312 [red]** such that feature sets 312 are linearly uncorrelated variables derived from the histogram of oriented gradient features.” *Id.*, 8:57-63. According to Chalom, each “feature set correspond[s] to each eye region” (*Id.*, 9:4-6) and therefore it is my opinion that the recited “configuration of the target feature of each input patch” is represented by such feature set.

292. Second, it is my opinion that Chalom-Aslan discloses that “the first and second selection instructions define how the feature vector is used to select editing instructions for the input patch” required by 2[B]. As I explained earlier in Section XII.C.2.e, the Chalom-Aslan system includes a trained teacher model and a trained student model, each of which receives input images and generates optimal motion vector fields based on the feature sets extracted from those images.

293. Therefore, in my opinion, Chalom discloses “the configuration of the target feature of each input patch is represented by a feature vector [e.g., Chalom’s feature set 312] derived from plural local descriptors [e.g., Chalom’s HOG features] of the input patch, and the first and second selection instructions [e.g., the trained teacher model and the trained student model in Chalom-Aslan system] define how the feature vector is used to select editing instructions [e.g., motion vector fields] for the input patch.”

4. Claim 3

294. In my opinion, claim 3 is unpatentable because it is rendered obvious by Chalom in view of Aslan. Claim 3 of the ’985 Patent depends from claim 1 and recites:

3. The method of claim 1, wherein the editing instructions comprise a displacement vector field defining how the input patch is to be transformed.

295. As I explained above in Section XII.A.3, the '985 Patent admits that the added feature for claim 3 is not new. Regardless, it is my opinion that the added feature for claim 3 is taught by Chalom in view of Aslan.

296. According to the '985 Patent, “a displacement vector field that represents the transformation of the image patch in the input image required to obtain the image patch of the output image.” Ex. 1001, 8:47-51. Specifically, “The displacement vector field F is chosen so that the image patch of the output image $O(x,y)$ is derived from the image patch of the input image $I(x,y)$ as

$$O(x,y)=I(x+u(x,y),y+v(x,y))”$$

Id., 8:58-62.

297. Like the '985 Patent, Chalom’s editing instructions comprise motion vector fields. *See* my discussion with respect to claim 1 in Section XII.C.2.

298. According to Chalom, motion vector fields “may adjust the input (e.g., having an uncorrected eye region) image to a desired output (e.g., having corrected eye contact)” by “warp[ing] the eye region.” Ex. 1005, 3:28-31, 3:36-38; 9:30-38 (“motion vector fields 313 may be applied to eye regions 311 by **determining, for each pixel location of eye regions 311 a warped pixel value** as a pixel value corresponding to pixel value as indicated by the motion vector for that pixel.”).

299. Therefore, in my opinion, Chalom-Aslan discloses “the editing instructions comprise a displacement vector field [e.g., Chalom’s motion vector field] defining how the input patch is to be transformed [e.g., how an eye region is warped].”

5. Claim 5

300. In my opinion, claim 5 is unpatentable because it is rendered obvious by Chalom in view of Aslan. Claim 5 of the ’985 Patent recites the following.

5[P] The method of claim 1, wherein:

5[A] a first editing algorithm is used by the first machine learning algorithm to define how the first editing instructions are to be applied to an input patch to derive an output patch; and

5[B] a second editing algorithm is used by the second machine learning algorithm to define how the second editing instructions are to be applied to an input patch to derive an output patch.

301. As explained in Section XII.C.2.e, both the teacher and student models in Chalom-Aslan generate “editing instructions,” e.g., motion vector fields. According to Chalom, “the eye region(s) of the source image may be warped based on the motion vector field(s) and integrated into the source image (e.g., into the remaining, non-eye region portion of the source image) to generate the eye contact corrected image.” Ex. 1005, 4:12-16. In my opinion, the algorithms responsible for warping the eye regions in the teacher and student models for Chalom-Aslan

based on the motion vector fields are the first and second editing algorithms required by claim 5.

302. Therefore, in my opinion, Chalom in view of Aslan discloses that “a first editing algorithm [e.g., Chalom’s algorithm to warp the eye regions] is used by the first machine learning algorithm [e.g., Chalom-Aslan’s teacher model] to define how the first editing instructions [e.g., motion vector fields generated by the teacher model] are to be applied to an input patch [e.g., eye regions to be warped] to derive an output patch [e.g., warped eye regions]; and a second editing algorithm [e.g., Chalom’s algorithm to warp the eye regions] is used by the second machine learning algorithm [e.g., Chalom-Aslan’s student model] to define how the second editing instructions [e.g., motion vector fields generated by the student model] are to be applied to an input patch [e.g., eye regions to be warped] to derive an output patch [e.g., warped eye regions].”

6. Claim 6

303. Claim 6 depends from claim 5 (disclosed by Chalom-Aslan for reasons provided above) and recites:

6[P] The method of claim 5, wherein:

6[A] the second editing instructions are principle component analysis components of a principle component analysis of the first editing instructions;
and

6[B] the second image editing algorithm is configured to transform the second editing instructions into the first editing instructions by inverse principle component analysis transform.

304. I note here that, although the '985 Patent claims and specification recite “**principle** component analysis,” to the extent this is not indefinite, I understand it to refer to “**principal** component analysis.”

305. It is my opinion that the added aspects for claim 6 concerning “principle component analysis” are taught by the Chalom-Aslan combination.

306. Chalom discloses using principal component analysis to compress the input to its MLA, e.g., RFC. As shown in Fig. 3 below, Chalom discloses a “histogram of oriented gradients features module 304 [green], which may generate or determine histogram of oriented gradient [HOG] features for each eye region of eye regions 311.” Ex. 1005, 8:30-34.

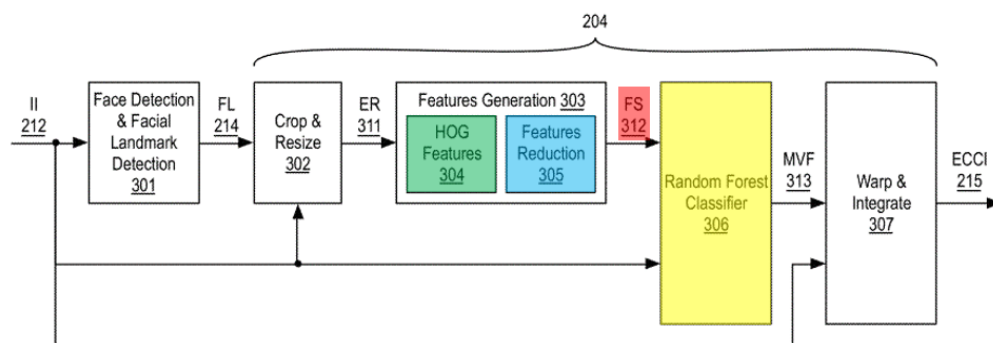


FIG. 3

Id., Fig. 3. These HOG features “may be reduced [by features reduction model 305 (blue)] using **principal component analysis** (PCA) techniques.” *Id.*, 8:57-59;

13:28-30. The reduced features are “feature sets 312 [red],” which “may be received by random forest classifier 306 [yellow].” *Id.*, 9:8-9. Such dimensionality reduction using PCA is also applied to HOG features generated from input patches during training. As shown in Fig. 8, Chalom discloses that “features determination module 805 [green] may generate histogram of oriented gradient (HOG) features and apply principal component analysis to reduce the number of features.” *Id.*, 13:11-14.

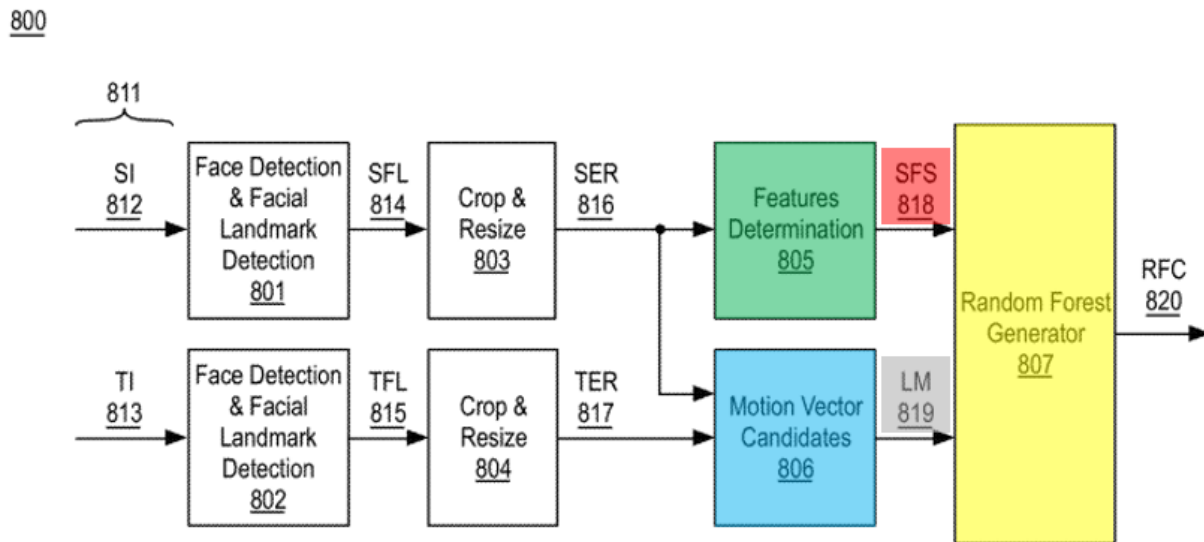


FIG. 8

Id., Fig. 8.

307. While Chalom discloses compressing one of the two inputs to its RFC, it is my opinion that a PHOSITA would have found it obvious to apply Chalom’s PCA to compress the other input to the teacher model, the input to the student model, or both. In other words, as explained above with respect to

Chalom's Fig. 3, Chalom teaches compressing one input to its MLA (e.g., RFC), it would have been obvious to a PHOSITA to use the same technique to compress other inputs to the MLAs in the Chalom-Aslan system to achieve a lighter and more efficient model. *See, e.g.*, Ex. 1007, ¶44. This is also consistent with Aslan's objective of efficiency. Specifically, when applying PCA to the student model's input, since the student takes both training data and "first editing instructions" as input, it is my opinion that a PHOSITA would have been motivated to apply PCA to the training data, the "first editing instructions," or both. It was well known long before the '985 Patent to apply PCA to different types of data, including flow fields. *See, e.g.*, Ex. 1035, p. 175; Ex. 1036, p. 1; *see also*, Ex. 1040, pp. 163, 179.

308. When applying PCA to compress the "first editing instructions," which are then used to train the student model, the trained student model would generate "second editing instructions" as the PCA components of the first editing instructions, and that such "second editing instructions" could be converted back to approximate the "first editing instructions" by inverse principal component analysis transform.

309. Therefore, in my opinion, Chalom in view of Aslan discloses claim 6.

7. Claim 8

310. Claim 8 depends from claim 1 (disclosed by Chalom-Aslan for reasons provided above) and recites:

8. The method of claim 1, wherein the first selection instructions for the first reference data are able to select between a larger number of alternative editing instructions than the second selection instructions for the second reference data.

311. In my opinion, the Chalom-Aslan combination discloses the added feature of “wherein the first selection instructions for the first reference data are able to select between a larger number of alternative editing instructions than the second selection instructions for the second reference data.”

312. As I explained in Section XII.C.1 above, Chalom-Aslan can include a teacher model being a complex RFC and a student model being a more compact RFC. A more compact RFC may have fewer trees and/or have each tree with fewer leaves. Since each leaf corresponds to editing instructions, the student RFC in the Chalom-Aslan system has fewer leaves, and therefore fewer editing instructions than the teacher RFC.

313. As I also explained in Section XII.C.1 above, Chalom-Aslan can also include a teacher model being a neural network and a student being a random forest. In this example implementation, the same logic applies. For example, it was well known that a neural network can be quite complex and therefore includes more weights and more representations of possible editing instructions, such as one million possible editing instructions. In contrast, the student RFC may have certain number of trees providing, for example, a thousand leaves representing a

thousand possible editing instructions. Thus, the student RFC in the Chalom-Aslan system provides fewer editing instructions than the teacher neural network.

314. In fact, this logic similarly applies when both the teacher and student models in Chalom-Aslan are neural networks, where the teacher model is a larger neural network providing more possible editing instructions and the student model is a smaller neural network providing fewer possible editing instructions. *See generally*, Section V.D.1.

315. Therefore, in my opinion, Chalom in view of Aslan discloses claim 8.

8. Claim 9

316. Claim 9 depends from claim 1 (disclosed by Chalom-Aslan for reasons provided above) and recites:

9. The method of claim 1, wherein the first machine learning algorithm is of a different machine learning algorithm type than the second machine learning algorithm.

317. In my opinion, the Chalom-Aslan combination discloses the added feature of the two claimed MLAs being of different types.

318. As I explained in Section XII.C.1 above, it would have been obvious for Chalom-Aslan to have the teacher model as a neural network and the student model as a random forest. *See, e.g.*, Ex. 1005, 9:13-17; Ex. 1007, ¶22.

319. Therefore, in my opinion, Chalom in view of Aslan discloses “the first machine learning algorithm [e.g., teacher model, such as a NN] is of a different machine learning algorithm type than the second machine learning algorithm [e.g., student model, such as a random forest].”

9. Claim 10

320. Claim 10 depends from claim 1 and recites:

10. The method of claim 1, wherein the first machine learning algorithm comprises one or more of the following: a neural network; a support vector machine; and a generative adversarial network (GAN).

321. As I just explained, in Chalom-Aslan, the first MLA can be a NN, so in my opinion, claim 10 is rendered obvious by Chalom in view of Aslan.

10. Claim 11

322. Claim 11 depends from claim 1 and recites:

11. The method of claim 1, wherein the second machine learning algorithm comprises one or more of the following: a regression forest; regression ferns, cluster centres, a lookup table, and separable filter banks.

323. As I just explained, in Chalom-Aslan, the second MLA can be a regression forest, so, in my opinion, claim 11 is rendered obvious by Chalom in view of Aslan.

11. Claim 12

324. Claim 12 depends from claim 1 (which is disclosed by Chalom-Aslan for reasons provided above) and recites:

12. The method of claim 1, wherein **the target feature comprises** one or more of the following: **an eye region comprising at least part of an eye**, a nose region comprising at least part of a nose, a mouth region comprising at least part of a mouth, a chin region comprising at least part of a chin, a neck region comprising at least part of a neck, and a hair region comprising hair.

325. Because Chalom is directed to eye gaze correction and Chalom’s “target feature comprises... an eye region comprising at least part of an eye,” in my opinion, claim 12’s additional limitation is rendered obvious by Chalom in view of Aslan for reasons I explained in Sections XII.C.2.c-XII.C.2.d.

12. Claim 13

326. In my opinion, claim 13 is unpatentable because it is rendered obvious by Chalom in view of Aslan. Claim 13 of the ’985 Patent recites the following. I address each of these in my analysis below.

13[P] A method of adjusting a digital representation of a head region, the method comprising

13[A] receiving a digital representation of a head region; and

13[B] using reference data comprising editing instructions to adjust the digital representation of the head region, wherein the reference data

comprises the second reference data generated by the method of claim 1.

a. 13[P]

327. To the extent that 13[P] is limiting, it focuses on runtime use of the reference data from claim 1.

328. Chalom discloses the same in its Figure 3, which teaches “[a] machine implemented **method** for providing **eye contact correction**.” Ex. 1005, claim 1; Fig. 3.

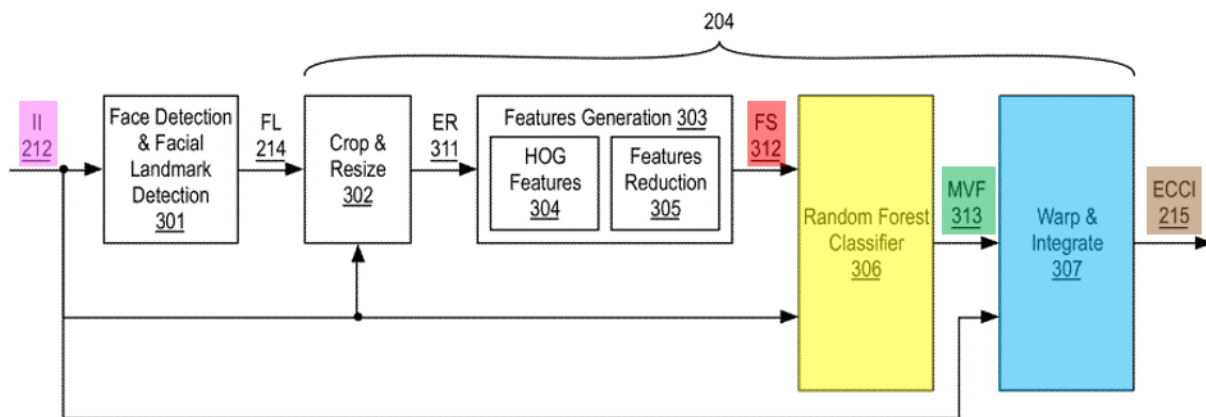


FIG. 3

Id., Fig. 3.

329. Therefore, in my opinion, Chalom discloses 13[P].

b. 13[A]

330. 13[A] recites “receiving a digital representation of a head region,” which, in my opinion, is disclosed by the Chalom-Aslan combination.

331. As shown in Fig. 3 below, Chalom discloses receiving input image 212 (pink).

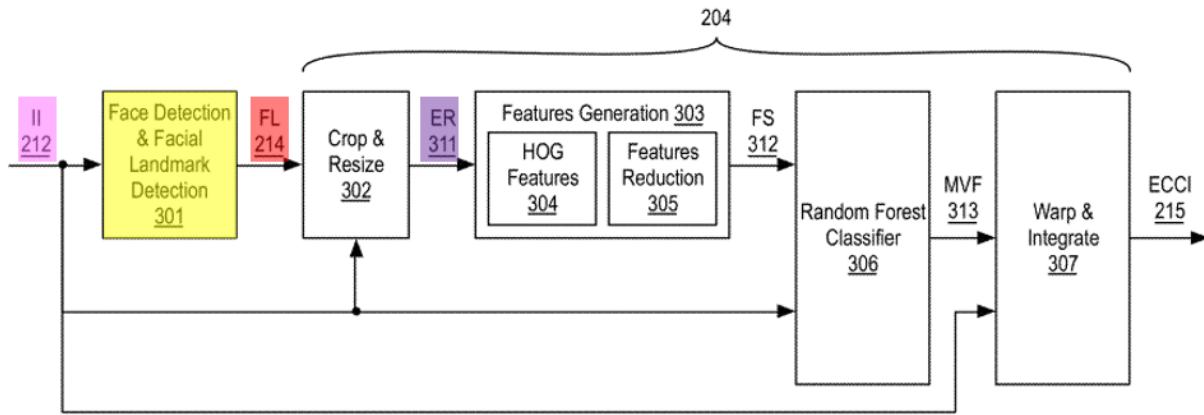


FIG. 3

Ex. 1005, Fig. 3; 7:48-52.

332. As I detailed above, Chalom's Fig. 3 also illustrates runtime use of trained RFC 306 in Chalom-Aslan, where Aslan's teacher-student architecture is adopted for training Chalom's MLA, e.g., RFC 306. *Supra* Section XII.C.1.

333. Therefore, in my opinion, Chalom-Aslan discloses "receiving a digital representation of a head region [e.g., input image 212 having a face]."

c. 13[B]

334. 13[B] recites "using reference data comprising editing instructions to adjust the digital representation of the head region, wherein the reference data comprises the second reference data generated by the method of claim 1," which, in my opinion, is disclosed by the Chalom-Aslan combination.

335. As I explained above, Chalom-Aslan uses a student model to generate motion vector fields during training. *Supra* Section XII.C.2.f. Thus, in my opinion, by using the “second selection instructions” (e.g., the trained student model, which is part of the “second reference data” generated during training), Chalom-Aslan discloses using the “second reference data” generated by the method of claim 1, i.e., generated **during training**, for gaze redirection **during runtime**.

336. Therefore, in my opinion, Chalom in view of Aslan discloses 13[B].

13. Claim 14

337. Claim 14 recites the following:

14. A non-transitory computer readable storage medium storing a computer program capable of execution by a processor and arranged on execution to cause the processor to perform a method according to claim 1.

338. In my opinion, claim 14 is disclosed by the Chalom-Aslan combination.

339. As shown in Fig. 16 below, Chalom discloses that “face detection module 202, facial landmark detection module 203, eye contact correction module 204, and video compression module 205 may be implemented via **central processor 1601** [yellow].” Ex. 1005, 15:47-50.

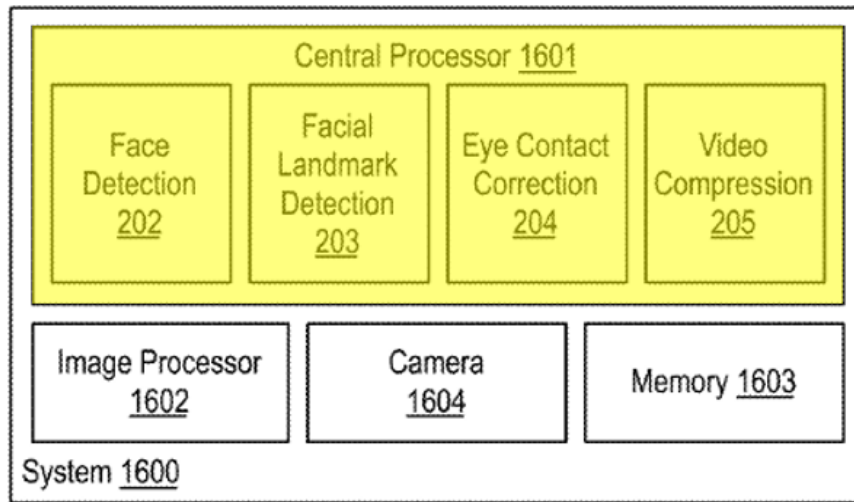


FIG. 16

Id., Fig. 16.

340. Aslan also discloses, in Figure 8 below, “an exemplary computing system environment 800 for implementing the joint training techniques and systems.” Ex. 1007, ¶65.

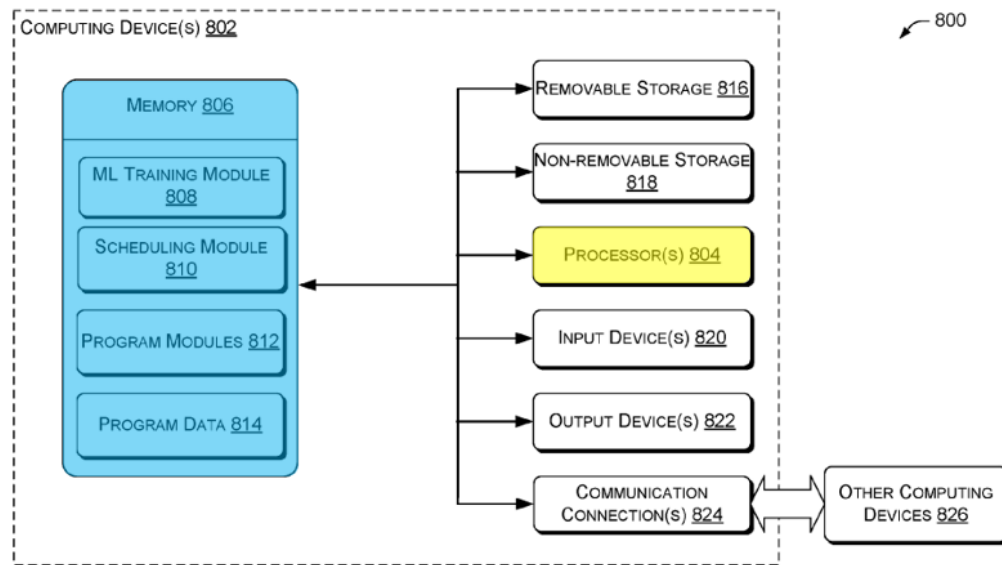


Fig. 8

Id., Fig. 8. As shown, Aslan discloses a computing device including “one or more processors 804 [yellow] and computer-readable memory 806 [blue].” *Id.*, ¶66.

“The processor(s) 804 can be configured to execute instructions, applications, or programs stored in the memory 806.” *Id.*

341. Therefore, it is my opinion that, in view of these teachings, it would have been obvious (and indeed trivial) to run the Chalom-Aslan teacher-student architected training process on a processor. Thus, claim 14 is obvious.

14. Claim 16

342. In my opinion, claim 16 is unpatentable because it is rendered obvious by Chalom in view of Aslan. Claim 16 of the '985 Patent recites the following. I address each of these in my analysis below.

16[P] A device for generating reference data for adjusting a digital representation of a head region, the device comprising a data processing unit arranged to:

16[A] receive training data comprising:

16[B] a set of input patches, each input patch comprising information about a target feature of a digital representation of the head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

16[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target portion of the digital representation of the head region after adjustment of the digital representation of the head region;

16[D] use a first machine learning algorithm to generate first reference data using the training data; and

16[E] use a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data output by the first machine learning algorithm

16[F] wherein the first reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

16[G] wherein the second reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region, and

16[H] wherein the second editing instructions in the second reference data are provided in a compressed representation.

a. 16[P]

343. As shown below, most of 16[P] is identical to 1[P] (identical portion indicated in bold).

Claim 16 Limitation		Claim 1 Limitation	
16[P]	A device for generating reference data for adjusting a digital representation of a head region , the device comprising a data processing unit arranged to:	1[P]	A method of generating reference data for adjusting a digital representation of a head region , the method comprising:

344. As to the trivial “device” and “processing unit” recitations, Chalom discloses in Fig. 14 “a flow diagram illustrating an example process 1400 for **pretraining an eye contact correction classifier.**” Ex. 1005, 13:66-67.

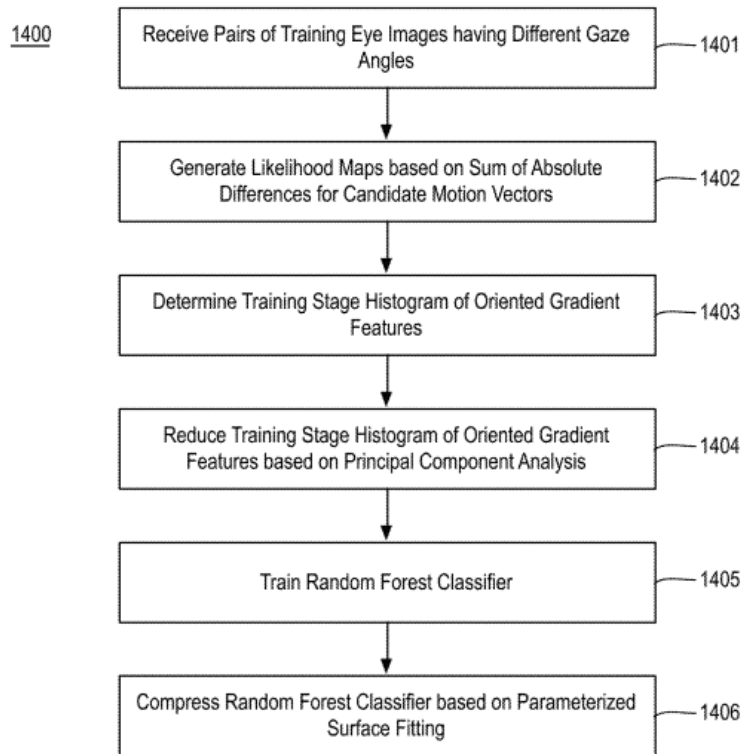


FIG. 14

Id., Fig. 14. “[P]rocess 1400 may be performed by system 1600” as shown in Fig. 16 below. *Id.*, 14:8-10.

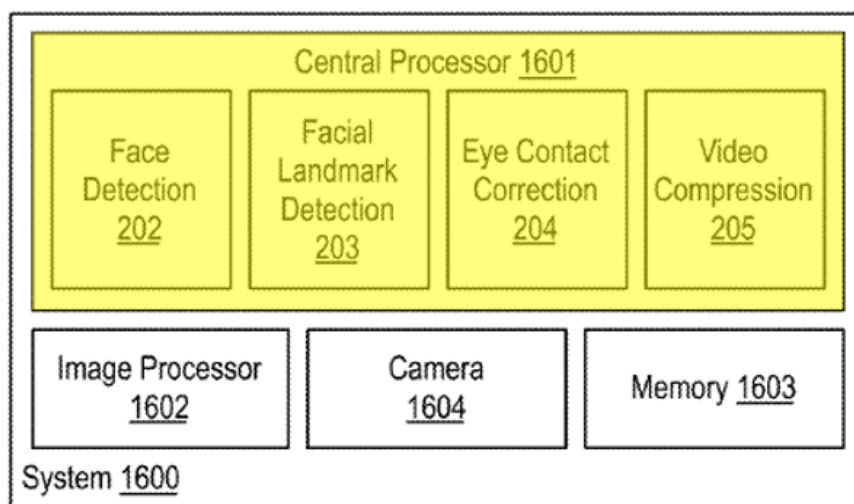


FIG. 16

Id., Fig. 16; 15:31-40.

345. In my opinion, in view of at least these teachings, it would have been obvious to have a data processing unit on a device arranged to perform the training steps of the combined Chalom-Aslan system. *See* Sections XII.C.2.aXII.C.13 above.

346. Therefore, in my opinion, the 16[P] is disclosed by Chalom in view of Aslan.

b. 16[A]-[C]

347. As I show below, 16[A-C] are substantively identical to 1[A-C] respectively.

Claim 16 Limitation	Claim 1 Limitation
---------------------	--------------------

16[A]	receive training data comprising:	1[A]	receiving training data comprising:
16[B]	a set of input patches, each input patch comprising information about a target feature of a digital representation of the head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and	1[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
16[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target portion of the digital representation of the head region after adjustment of the digital representation of the head region;	1[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

348. Therefore, in my opinion, 16[A-C] are disclosed by Chalom-Aslan for the same reasons I explained in Sections XII.C.2.b-XII.C.2.d.

c. 16[D] and 16[F]

349. As I show below, 16[D] and 16[F] are substantively identical to Limitation 1[D].

Claim 16 Limitation		Claim 1 Limitation	
16[D]	use a first machine learning	1[D]	using a first machine learning

	algorithm to generate first reference data using the training data; and		algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
16[F]	wherein the first reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,		

350. Therefore, in my opinion, 16[D] and 16[F] are disclosed by Chalom-Aslan for the same reasons I explained in Section XII.C.2.e.

d. 16[E] and 16[G]

351. As I show below, 16[E] and 16[G] are substantively identical to 1[E].

Claim 16 Limitation		Claim 1 Limitation	
16[E]	use a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data output by the first machine learning algorithm	1[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the
16[G]	wherein the second reference data comprises editing instructions for adjusting the digital representation of the head region for a range of possible digital representations		

	of the head region, and		head region,
--	-------------------------	--	--------------

352. Therefore, in my opinion, 16[E] and 16[G] are disclosed by Chalom-Aslan for the same reasons I explained in Section XII.C.2.e.

e. 16[H]

353. 16[H] recites that “the second editing instructions in the second reference data are provided in a compressed representation,” which, in my opinion, is disclosed by Chalom-Aslan.

354. According to the ’985 Patent, a compressed representation may be a principal component analysis representation. *See* Ex. 1001, claim 19.

355. As I explained earlier, the second editing instructions can be the PCA components of the first editing instructions, which means that they are provided in a compressed representation. *Supra* Section XII.C.6.

356. Therefore, in my opinion, 16[H] is disclosed by Chalom in view of Aslan.

15. Claim 17

357. In my opinion, claim 17 is unpatentable because it is rendered obvious by Chalom in view of Aslan. Claim 17 of the ’985 Patent recites the following. I address each of these in my analysis below.

17[P] A method of generating reference data for adjusting a digital representation of a head region, the method comprising:

17[A] receiving training data comprising:

17[B] a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

17[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

17[D] using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and

17[E] using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

17[F] wherein the first machine learning algorithm comprises a first neural network and the second machine learning algorithm comprises a second neural network, and wherein the second neural

network comprises fewer layers and/or smaller convolution fields than the first neural network.

a. 17[P] and 17[A-E]

358. As I show below, 17[P] and 17[A-E] are identical to 1[P] and 1[A-E], respectively.

Claim 17 Limitation		Claim 1 Limitation	
17[P]	A method of generating reference data for adjusting a digital representation of a head region, the method comprising:	1[P]	A method of generating reference data for adjusting a digital representation of a head region, the method comprising:
17[A]	receiving training data comprising:	1[A]	receiving training data comprising:
17[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and	1[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
17[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;	1[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

17[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and	1[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
17[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,	1[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

359. Therefore, in my opinion, 17[P] and 17[A-E] are disclosed for the same reasons I explained in Sections XII.C.2.a-XII.C.2.e.

b. 17[F]

360. 17[F] requires that “the first machine learning algorithm comprises a first neural network and the second machine learning algorithm comprises a second

neural network, and wherein the second neural network comprises fewer layers and/or smaller convolution fields than the first neural network,” which, in my opinion, is disclosed by Chalom-Aslan.

361. As I explained earlier, the teacher model in the Chalom-Aslan system could be a NN. *Supra* Section XII.C.1. While I discussed two example ways of implementing changes to Chalom in view of Aslan’s teachings in Section XII.C.1, it is my opinion that a PHOSITA would have found it obvious to implement in a third example way, i.e., the student model also being a NN. *See* Ex. 1005, 9:13-17 (“Although illustrated with respect to random forest classifier 306, **any suitable pretrained classifier** may be applied...”).

362. According to Aslan, the student model “can comprise a much smaller machine learning model (e.g., a neural net with **1000 times fewer parameters** than the first model 100) that has the **size and/or speed that is advantageous** at run-time in particular scenarios.” Ex. 1007, ¶44.

363. To achieve Aslan’s objective to convert a bigger but slower teacher NN into a smaller and faster student NN, it is my opinion that the student NN is expected to have fewer layers, smaller convolutional fields, or both.

16. Claim 18

364. In my opinion, claim 18 is unpatentable because it is rendered obvious by Chalom in view of Aslan. Claim 18 of the '985 Patent recites the following. I address each of these in my analysis below.

18[P] A method of generating reference data for adjusting a digital representation of a head region, the method comprising:

18[A] receiving training data comprising:

18[B] a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

18[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

18[D] using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and

18[E] using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head

region for a range of possible digital representations of the head region,

18[F] wherein the target feature comprises an eye region comprising at least part of an eye and the adjustment of the digital representation of the head region comprises adjusting a gaze direction, and/or

wherein the target feature comprises a nose region comprising at least part of a nose and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the nose, and/or

wherein the target feature comprises a chin region comprising at least part of a chin and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the chin, and/or

wherein the target feature comprises a neck region comprising at least part of a neck and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the neck, and/or

wherein the target feature comprises a hair region comprising hair and the adjustment of the digital representation of the head region comprises adjusting a color of the hair.

a. 18[P] and 18[A-E]

365. As I show below, 18[P] and 18[A-E] are identical to 1[P] and 1[A-E], respectively.

Claim 18 Limitation		Claim 1 Limitation	
18[P]	18. A method of generating	1[P]	A method of generating

	reference data for adjusting a digital representation of a head region, the method comprising:		reference data for adjusting a digital representation of a head region, the method comprising:
18[A]	receiving training data comprising:	1[A]	receiving training data comprising:
18[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and	1[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and
18[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;	1[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
18[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and	1[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and

18[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,	1[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,
-------	--	------	--

366. Therefore, in my opinion, 18[P] and 18[A-E] are disclosed for the same reasons I explained in Sections XII.C.2.a-XII.C.2.e.

b. 18[F]

367. 18[F] is substantially similar to an aspect in Claim 12: (identical portion indicated in bold)

Claim 18 Limitation		Claim 12 Limitation	
18[F]	wherein the target feature comprises an eye region comprising at least part of an eye and the adjustment of the digital representation of the head region comprises adjusting a gaze direction, and/or wherein the target feature	12	12. The method of claim 1, wherein the target feature comprises one or more of the following: an eye region comprising at least part of an eye , a nose region comprising at least part of a nose, a mouth region comprising at least part of a mouth, a chin region comprising at least part of a

	comprises a nose region comprising at least part of a nose and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the nose, and/or wherein the target feature comprises a chin region comprising at least part of a chin and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the chin, and/or wherein the target feature comprises a neck region comprising at least part of a neck and the adjustment of the digital representation of the head region comprises adjusting a shape and/or texture of the neck, and/or wherein the target feature comprises a hair region comprising hair and the adjustment of the digital representation of the head region comprises adjusting a color of the hair.		chin, a neck region comprising at least part of a neck, and a hair region comprising hair.
--	---	--	---

368. Therefore, in my opinion, for the same reasons I provided for in Section XII.C.11, 18[F] is also taught by Chalom in view of Aslan.

17. Claim 19

369. In my opinion, claim 19 is unpatentable because it is rendered obvious by Chalom in view of Aslan. Claim 19 of the '985 Patent recites the following. I address each of these in my analysis below.

19[P] A method of generating reference data for adjusting a digital representation of a head region, the method comprising:

19[A] receiving training data comprising:

19[B] a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

19[C] a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;

19[D] using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and

19[E] using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head

region for a range of possible digital representations of the head region,

19[F] wherein the second editing instructions in the second reference data are provided in a compressed representation, and

19[G] wherein the compressed representation comprises one or more of the following: a principle component analysis representation; a wavelet representation; Fourier and/or discrete cosine transform components; cluster centers.

a. 19[P] and 19[A-E]

370. As I show below, 19[P] and 19[A-E] are identical to 1[P] and 1[A-E], respectively.

Claim 19 Limitation		Claim 1 Limitation	
19[P]	19. A method of generating reference data for adjusting a digital representation of a head region, the method comprising:	1[P]	A method of generating reference data for adjusting a digital representation of a head region, the method comprising:
19[A]	receiving training data comprising:	1[A]	receiving training data comprising:
19[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and	1[B]	a set of input patches, each input patch comprising a target feature of a digital representation of a head region prior to adjustment of the digital representation of the head region, wherein the target feature is the same for each input patch; and

19[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;	1[C]	a set of output patches in one-to-one correspondence with the input patches, each output patch comprising the target feature of the digital representation of the head region after adjustment of the digital representation of the head region;
19[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and	1[D]	using a first machine learning algorithm to generate first reference data using the training data, the first reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region; and
19[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,	1[E]	using a second machine learning algorithm to generate second reference data using the same training data as the first machine learning algorithm and the first reference data generated by the first machine learning algorithm, the second reference data comprising editing instructions for adjusting the digital representation of the head region for a range of possible digital representations of the head region,

371. Therefore, in my opinion, 19[P] and 19[A-E] are disclosed by Chalom in view of Aslan for the same reasons I provided for in Sections XII.C.2.a-XII.C.2.e.

b. 19[F]

372. 19[F] recites “wherein the second editing instructions in the second reference data are provided in a compressed representation,” which is substantively identical to 16[H], which recites “wherein the second editing instructions in the second reference data are provided in a compressed representation.” Therefore, in my opinion, 19[F] is disclosed for the same reasons I explained in Section XII.C.14.e.

c. 19[G]

373. 19[G] recites, in part, that “the compressed representation comprises one or more of the following: a principle component analysis representation...,” which, in my opinion, is disclosed in Chalom-Aslan for reasons I explained in Section XII.C.6 above.

D. My Opinions as to Claim 4 based on Chalom in view of Aslan and Grau

1. Claim 4

374. In my opinion, claim 4 is unpatentable because it is rendered obvious by Chalom in view of Aslan and further in view of Grau. Claim 4 of the '985 Patent depends from claim 1 and recites:

4. The method of claim 1, wherein the editing instructions comprise a filter field, a brightness adjustment field, or a texture blending field.

375. In my opinion, the added feature requiring that “the editing instructions comprise a filter field, a brightness adjustment field, or a texture blending field” is disclosed in Grau.

376. According to the '985 Patent, “[t]he editing instructions are not limited to displacement vector fields” and may “additionally or alternatively” be “a filter field, brightness adjustment field, or texture blending field.” Ex. 1001, 9:13-29 (“filter field”), 9:37-38 (“brightness adjustment field”), 9:35-37 (“texture blending field”).

377. In my opinion, while Chalom discloses “displacement vector fields” as “editing instructions” (*supra* Section XII.C.4), Grau discloses two of the other three: a “filter field” and a “brightness adjustment field” in the context of IR-to-color mapping and IR-and-lighting mapping.

378. Regarding IR-to-color mapping, Grau discloses that “the IR to color conversion may be performed by a mapping function **using a limited neighborhood of pixels (such as 1, 4, 8, higher order) to form a combination value (RGB)** for a single color value for a pixel.” Ex. 1009, ¶79. A PHOSITA would have understood that using limited neighborhood pixels to determine a pixel value is a filtering operation, and that the mask used for performing such filtering

operation is a filter field. In fact, the term “filter” in image processing is quite broad and covers various types of local region pixel manipulation. *See, e.g.*, Ex. 1047, pp. 185-186 (“filter... 4. In computer graphics, a special effect or production effect that is applied to bitmapped images; for example, shifting pixels within an image, making elements of the image transparent, or distorting the image.”).

379. Regarding the IR-and-lighting mapping, Grau uses “a neural network such as a convolutional neural network (CNN)... to map the IR [infa-red] and lighting from non-occluded parts of the face.” Ex. 1009, ¶80. “Neural networks (like CNNs) can take multiple inputs, ...and maps [*sic*] this to the anticipated facial image.” *Id.* Changing the lighting/brightness of the occluded parts of the face to match those of the non-occluded part of the face is a lighting adjustment operation, and the lighting/brightness adjustment amount is a “brightness adjustment field.”

380. Therefore, in my opinion, Chalom in view of Aslan and further in view of Grau discloses claim 4.

2. Motivation to Combine Chalom-Aslan with Grau

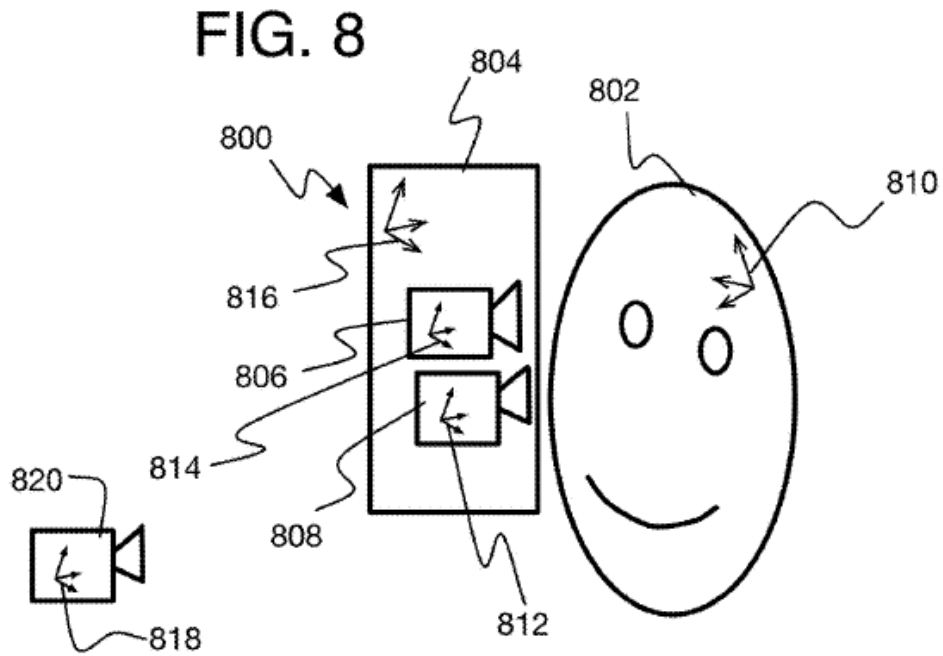
381. In my opinion, a PHOSITA would have been motivated to modify Chalom-Aslan based on Grau by replacing or supplementing Chalom’s training data, such that the teacher model and/or the student model generate a filter field, a brightness adjustment field, or a texture blending field, as opposed to the original motion vector fields.

382. Grau, like Chalom and Aslan, qualifies as analogous art to the '985 patent as it is in the **same field of endeavor**, *i.e.*, using machine learning for image processing. Ex. 1009, ¶79; Ex. 1001, Abstract; Ex. 1005, 3:21-32; Ex. 1007, ¶24. Chalom-Aslan, Grau, and the '985 Patent all use training data for training an MLA to perform facial image editing. *See, e.g.*, Ex. 1001, 1:11-12 (“correct a perceived gaze direction”); Ex. 1005, Title (“eye contact correction”); Ex. 1009, ¶79 (“IR to color conversion”), ¶80 (“IR and lighting” mapping), Fig. 2A (showing a rendered “image of an eye from a representation in a virtual or augmented reality”), ¶28 (“eyes can be rendered in real-time”), ¶42 (“the parameters for warping all can be obtained from the IR images such as eye gaze points”), ¶43 (“performed for multiple head poses, eye gaze directions, and facial expressions”). At a minimum, Grau is reasonably pertinent to at least one problem identified by the '985 Patent.

383. Although Chalom discloses a “warp and integrate module 307,” it does not provide much detail regarding how “the warped eye regions may be integrated into the remaining portion of input image.” Ex. 1005, 9:26-38. In my opinion, a PHOSITA seeking to integrate with high quality the warped eye regions would have been motivated to look to analogous references, such as Grau, for more detailed teachings on how to achieve this, such as by “mapping IR and lighting from non-occluded parts of the face by using convolutional neural networks (CNNs).” Ex. 1009, ¶42. In my opinion, a PHOSITA would also have been

motivated to apply the Chalom-Aslan system to additional image-optimization tasks that complement the basic gaze-correction task. It was well known that high-quality gaze correction often involves texture blending, which often includes filtering and brightness adjustment. *See, e.g.*, Ex. 1037, Ex. 1038, Ex. 1039, Ex. 1058. Chalom explicitly motivates a PHOSITA to employ its “techniques and/or arrangements... in a variety of other systems and applications other than what is described.” Ex. 1005, 2:11-17, 24:1-5 (“Hence, various modifications of the implementations described herein, as well as other implementations, which are apparent to persons skilled in the art to which the present disclosure pertains are deemed to lie within the spirit and scope of the present disclosure.”).

384. Moreover, Grau’s HMD has “one or more internal cameras pointed toward the face of the user 304, and particularly the eyes and area of the face around the eyes.” Ex. 1009, ¶34.



Ex. 1009, Fig. 8. This setup is analogous to Fig. 1 of the '985 Patent, where “a stereoscopic pair of images of a head is captured by the device 10.” Ex. 1001, 4:19-21.

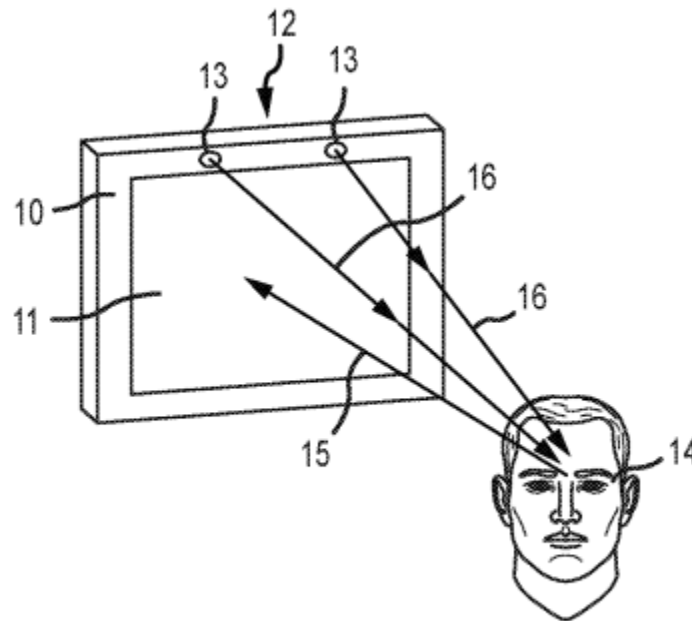


FIG.1

Id., Fig. 1.

385. Thus, in my opinion, a PHOSITA would have been motivated to expand the application of Chalom-Aslan and would have looked to Grau for specific facial-image editing instructions beyond displacement vector fields or even beyond gaze correction.

386. Finally, it is my opinion that a PHOSITA would have had a **reasonable expectation of success** in using the Chalom-Aslan system to perform the facial-image editing tasks described in Grau. Like the gaze correction in the Chalom-Aslan system, Grau’s “IR to color conversion” and “IR and lighting” mapping are also implemented via machine learning. Ex. 1009, ¶79 (“the IR to color conversion may be performed by a mapping function” and “[t]he mapping

can be implemented by means (averages) codebooks or other machine learning methods.”); ¶80(“a neural network, such as a convolutional neural network (CNN) may be used to map the IR and lighting from non-occluded parts of the face.”). A PHOSITA would have understood that applying Chalom-Aslan to Grau’s facial image editing merely involves modifying Chalom’s warping algorithm to incorporate editing instructions that are generated by Chalom-Aslan in view of Grau’s facial image editing; training data could still be the same, i.e., consisting of the disclosed “set of input patches” and “set of output patches in one-to-one correspondence with the input patches” and the objective functions being optimized could stay the same (i.e., minimizing the difference between the edited input patches and the corresponding ground-truth output patches and/or minimizing a difference between the “first reference data” of MLA1 and “the second reference data” of MLA2).

387. To the extent that more robust “IR and lighting” mapping is expected, Grau discloses the type of training data that can be used—“a training set that contains many examples of different persons under different lighting conditions.” Ex. 1009, ¶80. Grau also discloses that the training data can contain “different poses, facial expressions, and/or eye gaze directions.” Ex. 1009, ¶71. Thus, in my opinion, it would have been straightforward for a PHOSITA to implement Grau’s teachings into Chalom-Aslan.

E. My Opinions as to Claim 7 in view of Chalom-Aslan and Pace

1. Claim 7

388. Claim 7 depends from claim 5 (disclosed by Chalom-Aslan for reasons provided above) and recites:

7[P] The method of claim 5, wherein:

7[A] the second editing instructions are wavelet components of the first editing instructions; and

7[B] the second editing algorithm is configured to transform the second editing instructions into the first editing instructions by inverse wavelet transform.

389. Claim 7 is like claim 6 except wavelet transformation is used as opposed to principal component analysis.

390. Pace discloses using discrete wavelet transform (DWT) as a “[d]ata reduction technique” used to make “the input data... sparser.” Ex. 1010, 21:4-13. This is “because data is often more sparse in the wavelet domain” and “DWT transformation is the least ‘invasive’ and theoretically allows full recovery of the input data.” *Id.*, 21:6-13. It is my opinion that a PHOSITA would have found it obvious to use wavelet transformation for compression purposes of the editing instructions in Chalom-Aslan.

391. Therefore, in my opinion, claim 7 is disclosed by Chalom in view of Aslan and further in view of Pace.

2. Motivation to Combine Chalom-Aslan with Pace

392. In my opinion, a PHOSITA would have been motivated to modify Chalom-Aslan to use wavelet transformation, as explained below.

393. Pace, like Chalom and Aslan, qualifies as analogous art to the '985 Patent because it is in the **same field of endeavor**. Ex. 1010, 21:4-11, Ex. 1001, Abstract; Ex. 1005, 3:21-32; Ex. 1007, ¶24. Like the Chalom-Aslan system and the '985 Patent, Pace also utilizes data compression techniques to compress data to be used in a next operation. Ex. 1005, 8:49-9:7; Ex. 1001, 18:7-20; Ex. 1010, 21:4-11. At a minimum, it is my opinion that Pace is at least reasonably pertinent to a problem identified in the '985 Patent. Ex. 1010, 1:45-2:2.

394. In my opinion, while Chalom-Aslan utilizes PCA to compress the “first editing instructions,” a PHOSITA would have found it obvious to replace PCA with wavelet transformation. For example, Chalom teaches that “[t]he histogram of oriented gradient features may be reduced **using any suitable technique or techniques**,” and that PCA is just one example. Ex. 1005, 8:51-53. Thus, it is my opinion that a PHOSITA would have been motivated to look to other references, such as Pace, where different compression techniques are used.

395. In addition, both PCA and DWT are commonly-known data-compression techniques and were widely used as alternatives before the '985 Patent. *See* Ex. 1040, pp. 163-164 (PCA is also known as KL transform); Ex.

1041. Pace itself discusses both PCA and DWT. Ex. 1010, 17:49-52, 20:26-29.

“Of the data reduction techniques, DWT transformation is the least ‘invasive’ and theoretically allows full recovery of the input data.” Ex. 1010, 21:11-13.

Moreover, Chalom notes that PCA is a machine learning process and requires training. Ex. 1005, 8:64-9:1. In contrast, DWT is a deterministic process and does not involve training (and therefore easier to implement). Thus, it is my opinion that a PHOSITA would have been motivated to replace the PCA compression in the Chalom-Aslan system with DWT, which is a less invasive data compression technique to be implemented more easily.

396. Finally, it is also my opinion that a PHOSITA would have had a reasonable expectation of success in replacing Chalom-Aslan’s PCA with Pace’s DWT. As I mentioned above, PCA and DWT were widely used as alternatives and using one or the other would have been well within a PHOSITA’s capabilities. In addition, since DWT does not require training, replacing PCA with DWT simplifies compression calculations.

397. Therefore, in my opinion, Chalom in view of Aslan and further in view of Pace discloses Claim 7.

XIII. CONCLUDING STATEMENTS

398. In my opinion, the challenged claims 1-3, 5, 10, 12-14, and 18 are anticipated by Ganin. As such, in my opinion, these claims should be found unpatentable and cancelled.

399. In my opinion, claims 16 and 19 are obvious by Ganin in view of Liu. As such, in my opinion, these claims should be found unpatentable and cancelled.

400. In my opinion, the challenged claims 1-3, 5-6, 8-14, and 16-19 are rendered obvious by the Chalom-Aslan combination. As such, in my opinion, these claims should be found unpatentable and cancelled.

401. In my opinion, the challenged claim 4 is rendered obvious by the Chalom-Aslan combination further in view of Grau. As such, in my opinion, this claim should be found unpatentable and cancelled.

402. In my opinion, the challenged claim 7 is rendered obvious by the Chalom-Aslan combination further in view of Pace. As such, in my opinion, this claim should be found unpatentable and cancelled.

403. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code

and that such willful false statements may jeopardize the results of these proceedings.

Date: June 16, 2023



Lina J. Karam, Ph.D.