

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Axon Enterprise, Inc., Dedrone Holdings, Inc., and Skydio, Inc.

Petitioners,

v.

Airspace Systems, Inc.

Patent Owner

Case No. IPR2025-00629

U.S. Patent No. 10,514,711

**SECOND DECLARATION OF NIKOLAOS PAPANIKOLOPOULOS, PH.D.
IN SUPPORT OF PATENT OWNER’S REPLY TO
PETITIONERS’ OPPOSITION TO
PATENT OWNER’S REVISED MOTION TO AMEND**

TABLE OF CONTENTS

I. INTRODUCTION	18
II. COMPENSATION	19
III. MATERIALS CONSIDERED	20
IV. LEGAL PRINCIPLES.....	21
<i>A. Enablement and Written Description</i>	<i>21</i>
<i>B. Enablement of Prior Art</i>	<i>23</i>
V. SUMMARY OF MY OPINIONS.....	24
VI. THE '711 PATENT – INNOVATIVE ASPECTS OF PROPOSED SUBSTITUTE CLAIMS 21, 36, AND 27, AND CLAIM 12	27
<i>A. Overview.....</i>	<i>27</i>
<i>B. Drone's On-Board Use of Machine Learning Models / Neural Networks</i>	<i>40</i>
1. Challenges in Drone's On-Board Use of ML Models / Neural Networks	41
2. '711 Patent's Use of ML Models / Neural Networks On-Board a Drone	45
VII. SPECIFICATION OF THE '711 PATENT FULLY SUPPORTS ALL PROPOSED SUBSTITUTE CLAIMS	48
<i>A. Specification Fully Supports Proposed Substitute Claim 37</i>	<i>48</i>
1. Specification Discloses Image-by-Image Selection of a Neural Network	48
2. Specification Discloses Thresholds and Ranges	59
3. Specification Discloses Applying Second Neural Network When Confidence Is Higher than Second Threshold	64
<i>B. Specification Fully Supports SCs 21 and 24</i>	<i>65</i>
1. Written Description Supports Identifying a Model Type Based on Target Classification.....	65

2.	Written Description Teaches Two Speed Adjustment Factors for SCs 21 and 25, Respectively	67
VIII.	PSCs 21-37 ARE PATENTABLE	72
<i>A.</i>	<i>SCs 21-37 Are Enabled by the Specification</i>	<i>72</i>
1.	Claimed Speed Adjustment Factors Are Enabled	72
2.	Claim Limitations Robustness, Compute Cost, and Refresh Rate Are Enabled	79
3.	Claim Limitations First Threshold, Second Threshold, and Range Are Enabled	85
<i>B.</i>	<i>SCs 21-37 Are Not Indefinite</i>	<i>88</i>
1.	Terms Common to SCs 21, 36, and 37 Are Not Indefinite	88
2.	Limitations Recited in SC 37 Are Not Indefinite	91
<i>C.</i>	<i>Ground 23 – Beard, Laufer, Pierce, Khan, and Kumar (the “BLPKK combination”) Does Not Render SCs 21-27, 29, 31-33, and 35 Obvious..</i>	<i>95</i>
1.	A POSITA Would Not Combine BLPKK	95
2.	The BLPKK Combinations Does Not Teach or Suggest Each and Every Limitation of PSC 21	115
<i>D.</i>	<i>Grounds 24-26 Do Not Render SCs 28, 30, and 34 Obvious</i>	<i>122</i>
<i>E.</i>	<i>Ground 27 – Beard, Laufer, Pierce, and Ning (the BLPN Combination”) Does Not Render SC 36 Obvious</i>	<i>122</i>
1.	A POSITA Would Not Combine BLPN	122
2.	The BLPN Combination Does Not Teach or Suggest Each and Every Limitation of SC 36	136
<i>F.</i>	<i>Ground 28 –Pierce, Kumar, Khan, and Beard (the PKKB Combination”) Does Not Render SCs 21, 23-26, and 32 Obvious</i>	<i>159</i>
1.	A POSITA Would Not Have Combined or Modified PKKB	159
2.	The PKKB Combination Does Not Teach or Suggest Each and Every Limitation of SC 21	161
<i>G.</i>	<i>Grounds 29 and 30 – Further Combinations with Laufer and Hogasten Do Not Render SCs 22, 29, 34, and 35 Obvious</i>	<i>162</i>
<i>H.</i>	<i>Ground 31 –Pierce, Kumar, Ning, and Beard (the “PKNB Combination”) Does Not Render SC 36 Obvious</i>	<i>163</i>

1.	A POSITA Would Not Have Combined PKNB	163
2.	The PKNB Combination Does Not Teach or Suggest Each and Every Limitation of SC 36.....	164
I.	<i>Grounds 32-34 – Further Combinations with Sahasrabudhe, Laufer, and Hogasten Do Not Render SCs 21-26, 29, and 31-34 Obvious</i>	<i>164</i>
J.	<i>Ground 35 – Pierce, Kumar, Sahasrabudhe, and Ning (the “PKSN Combination”) Does Not Render SC 36 Obvious</i>	<i>165</i>
IX.	CONCLUSION.....	165

List of Patent Owner Exhibits

Exhibit No.	Description
EX2001	U.S. Army Contract No. W15QKN-17-9-1022
EX2002	Smithsonian, National Air and Space Museum Letter to Mr. Banga dated March 13, 2024
EX2003	Z. Yang and R. Sinnott, “Snake Detection and Classification using Deep Learning,” in Proceedings of Hawaii International Conference on System Sciences (2021), https://scholarspace.manoa.hawaii.edu/server/api/core/bitstreams/f9f2de4d-ddaa-4984-9fc3-9bbca4e27247/content
EX2004	Q. Ji, “Object Detection and Recognition,” https://sites.ecse.rpi.edu/~qji/CV/ObjectDetRecog2.pdf
EX2005	The National Science Foundation, “How to Radars Work?,” https://www.eol.ucar.edu/content/how-do-radars-work
EX2006	Principles of Doppler Radar, http://ww2010.atmos.uiuc.edu/(Gh)/guides/rs/rad/basics/dpmod.rxml
EX2007	https://developer.apple.com/documentation/corelocation/configuring-your-app-to-use-location-services
EX2008	https://developer.apple.com/documentation/corelocation/cllocation
EX2009	https://developer.android.com/develop/sensors-and-location/location/retrieve-current
EX2010	https://developer.android.com/reference/android/location/Location
EX2011	https://www.merriam-webster.com/dictionary/autonomous

Exhibit No.	Description
EX2012	https://dictionary.cambridge.org/us/dictionary/english/-autonomous
EX2013	Declaration of Noah Moore
EX2014	Airspace Interceptor, Fully Autonomous Physical Mitigation System Designed to Detect, Capture or Ram Drone, https://www.airspace.co/airspace-interceptor/
EX2015	Smithsonian National Air and Space Museum, Donate a Personal Item, https://airandspace.si.edu/donate-item
EX2016	Smithsonian National Air and Space Museum, Roger Connor Bio, https://airandspace.si.edu/people/staff/roger-connor
EX2017	Smithsonian National Air and Space Museum, About the Collection, available at https://airandspace.si.edu/collections/about-collection (last visited July 14, 2025)
EX2018	Jointly Proposed Protective Order
EX2019	U.S. Patent App. No. 16/142,452 (from which U.S. Patent No. 10,514,711 issued), as filed
EX2020	Streaming, Definition of Streaming by Merriam-Webster (2016)
EX2021	Streaming definition and meaning, Collins English Dictionary (2018)
EX2022	R. Wang et al., “An Effective Image Denoising Method for UAV Images via Improved Generative Adversarial Networks,” <i>Sensors</i> vol. 18(7):1985 (2018), doi:10.3390/s18071985.
EX2023	A. Ahmad et al., “Noise and Restoration of UAV Remote Sensing Images.” <i>International Journal of Advanced Computer Science</i>

Exhibit No.	Description
	<i>and Applications</i> , vol. 11, no. 12 (2020) doi:10.14569/ijacsa.2020.0111222
EX2024	Curriculum Vitae of Dr. Nikolaos Papanikolopoulos
EX2025	Declaration of Nikolaos Papanikolopoulos, Ph.D. in Support of Patent Owner's Response and First Motion to Amend
EX2026 (Served; Not filed)	Frequently Asked Questions, About the Internet Archive Affidavit and the Wayback Machine, https://archive.org/legal/faq
EX2027 (Served; Not filed)	Supplemental Declaration of Nikolos Papaniklopoulos, Ph.D., in support of Patent Owner's Response and Motion to Amend
EX2028	<i>Number not used</i>
EX2029	<i>Number not used</i>
EX2030	<i>Number not used</i>
EX2031	Withdrawn
EX2032	Withdrawn
EX2033	Transcript of Deposition of Dr. Singhose (Apr. 15, 2026)
EX2034	E. Biediger and W. Singhose, "Formation Flying Satellite Control Utilizing Input Shaping," in AIAA/AAS Astrodynamics Specialist Conf., Big Sky, Montana (2003).
EX2035	Withdrawn
EX2036	Declaration of Nikolaos Papanikolopoulos, Ph.D. in Support of Reply to Opposition to Revised Motion to Amend
EX2037	Declaration of June Munford
EX2038	Curriculum Vitae of June Munford

Exhibit No.	Description
EX2039	Vanhoucke, Vincent, Andrew Senior, and Mark Z. Mao. "Improving the speed of neural networks on CPUs." <i>Proc. deep learning and unsupervised feature learning NIPS workshop</i>. Vol. 1. (2011)
EX2040	Intel, Intel® Instruction Set Extensions Technology, Support Knowledge Base, <i>available at</i> https://www.intel.com/content/www/us/en/support/articles/000005779/processors.html
EX2041	Raman, Srinivas K., Vladimir Pentkovski, and Jagannath Keshava. "Implementing streaming SIMD extensions on the Pentium III processor." <i>IEEE micro</i> 20.4 (2000): 47-57.
EX2042	Suleiman, Amr, et al. "Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision." arXiv:1703.05853, arXiv, 17 Mar. 2017. <i>arXiv.org</i>, https://doi.org/10.48550/arXiv.1703.0585
EX2043	Ullah, Md Bahar. "CPU based YOLO: A real time object detection algorithm." <i>2020 IEEE region 10 symposium (TENSYP)</i>. IEEE (2020).
EX2044	Holleman, Matthijs, Real-time object detection with YOLO (2017), <i>available at</i> https://machinethink.net/blog/object-detection-with-yolo/
EX2045	Fang, Wei, Lin Wang, and Peiming Ren. "Tinier-YOLO: A real-time object detection method for constrained environments." <i>Ieee Access</i> 8 (2019): 1935-1944.

Exhibit No.	Description
EX2046	Ren, Shaoqing, et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks." arXiv:1506.01497, arXiv, 6 Jan. 2016. <i>arXiv.org</i> , https://doi.org/10.48550/arXiv.1506.01497 .
EX2047	Yang, Tianyu, and Antoni B. Chan. "Recurrent filter learning for visual tracking." <i>Proceedings of the IEEE international conference on computer vision workshops</i> (2017).
EX2048	Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." <i>The journal of machine learning research</i> 15.1 (2014): 1929-1958.
EX2049	Nakano, Reiichiro Christian S., et al. "Implementation of an artificial neural network in recognizing in-flight quadrotor images." <i>TENCON 2015-2015 IEEE Region 10 Conference</i> . IEEE (2015).
EX2050	Zhengj, Zejia, and Juyang Weng. "Mobile device based outdoor navigation with on-line learning neural network: A comparison with convolutional neural network." <i>Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops</i> (2016).
EX2051	Wong, Alexander, et al. "Yolo nano: A highly compact you only look once convolutional neural network for object detection. arxiv 2019." <i>arXiv preprint arXiv:1910.01271</i> 10 (2019).
EX2052	Apte, Maneesh, Simar Mangat, and Priyanka Sekhar. "YOLO Net on iOS." <i>CS231n. Stanford University</i> (2017).

Exhibit No.	Description
EX2053	Jain, Vaibhav, and Dinesh Patel. "A GPU based implementation of robust face detection system." <i>Procedia Computer Science</i> 87 (2016): 156-163.
EX2054	Lu, Yongyi, Cewu Lu, and Chi-Keung Tang. "Online video object detection using association LSTM." <i>Proceedings of the IEEE international conference on computer vision</i> (2017).
EX2055	Apaza-Agüero, Karl, Luciano Silva, and O. R. Bellon. "Mesh segmentation with connecting parts for 3D object prototyping." <i>2015 IEEE International Conference on Image Processing (ICIP)</i> (2015).
EX2056	Watanabe, Kouki, and Alexander G. Belyaev. "Detection of salient curvature features on polygonal surfaces." <i>Computer Graphics Forum</i> . Vol. 20. No. 3. Oxford, UK and Boston, USA: Blackwell Publishers Ltd (2001).
EX2057	Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." <i>2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)</i> . Vol. 1. IEEE (2005).
EX2058	Nvidia, GeForce GTX Titan X Specifications, <i>available at</i> https://www.nvidia.com/en-us/geforce/graphics-cards/geforce-gtx-titan-x/specifications/
EX2059	Nvidia, CUDA, https://www.nvidia.com/en-us/geforce/technologies/cuda/
EX2060	Excerpts of Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. <i>Foundations of machine learning</i> . MIT press (2018).

Exhibit No.	Description
EX2061	Redmon, Joseph, et al. "You Only Look Once: Unified, Real-Time Object Detection." arXiv:1506.02640, arXiv, 9 May 2016. <i>arXiv.org</i> , https://doi.org/10.48550/arXiv.1506.02640 .
EX2062	Iandola, Forrest N., et al. "SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size." arXiv:1602.07360, arXiv, 4 Nov. 2016. <i>arXiv.org</i> , https://doi.org/10.48550/arXiv.1602.07360 .
EX2063	Vanhoucke, Vincent, Andrew Senior, and Mark Z. Mao. "Improving the speed of neural networks on CPUs.", <i>available at</i> https://web.archive.org/web/20120130220632/http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/37631.pdf .
EX2064	Intel, Intel® Instruction Set Extensions Technology, Support Knowledge Base, <i>available at</i> https://web.archive.org/web/20190119075515/https://www.intel.com/content/www/us/en/support/articles/000005779/processors.html
EX2065	A library catalog MARC record for <i>IEEE Micro</i> , <i>available from Penn State University Library at</i> https://catalog.libraries.psu.edu/catalog/51599008 .
EX2066	Raman, Srinivas K., Vladimir Pentkovski, and Jagannath Keshava. "Implementing streaming SIMD extensions on the Pentium III processor." <i>IEEE Micro</i> 20.4 (2000): 47-57, <i>as sourced from the Penn State University Library's collection</i> .

Exhibit No.	Description
EX2067	A library catalog MARC record for <i>Proceedings of 2017 International Symposium on Circuits and Systems</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/22871497
EX2068	Suleiman, Amr, et al. "Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision," <i>Proceedings of 2017 International Symposium on Circuits and Systems</i> , as sourced from the Penn State University Library's collection.
EX2069	A library catalog MARC record for <i>Proceedings of the 2020 IEEE Region 10 Symposium</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/32192556
EX2070	Ullah, Md Bahar. "CPU based YOLO: A real time object detection algorithm." <i>2020 IEEE region 10 symposium (TENSYP)</i> . IEEE (2020), as sourced from the Penn State University Library's collection.
EX2071	Holleman, Matthijs, Real-time object detection with YOLO (2017), available at https://web.archive.org/web/20170526072201/https://machinethink.net/blog/object-detection-with-yolo/
EX2072	A library catalog MARC record for <i>IEEE Access</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/51598571
EX2073	Fang, Wei, Lin Wang, and Peiming Ren. "Tinier-YOLO: A real-time object detection method for constrained environments." <i>Ieee</i>

Exhibit No.	Description
	Access 8 (2019): 1935-1944, as sourced from the Penn State University Library's collection.
EX2074	A library catalog MARC record for <i>IEEE Transactions on Pattern Analysis and Machine Intelligence</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/51599284
EX2075	Ren, Shaoqing, et al. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," <i>IEEE Transactions on Pattern Analysis and Machine Intelligence</i> , as sourced from the Penn State University Library's collection.
EX2076	A library catalog MARC record for <i>Proceedings of the IEEE International Conference on Computer Vision Workshops</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/51598764
EX2077	Yang, Tianyu, and Antoni B. Chan. "Recurrent filter learning for visual tracking." Proceedings of the IEEE international conference on computer vision workshops (2017), as sourced from the Penn State University Library's collection.
EX2078	A library catalog MARC record for <i>Journal of Machine Learning Research</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/51620909
EX2079	Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958, as sourced from the Penn State University Library's collection.

Exhibit No.	Description
EX2080	A library catalog MARC record for <i>Proceedings of the IEEE Region Ten Conference</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/51718430
EX2081	Nakano, Reiichiro Christian S., et al. "Implementation of an artificial neural network in recognizing in-flight quadrotor images." TENCON 2015-2015 IEEE Region 10 Conference. IEEE (2015), <i>as sourced from the Penn State University Library's collection.</i>
EX2082	A library catalog MARC record for <i>Proceedings of the IEEE Conference on Computer Vision Workshops</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/51598764
EX2083	Zheng, Zejia, and Juyang Weng. "Mobile device based outdoor navigation with on-line learning neural network: A comparison with convolutional neural network." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (2016), <i>as sourced from the Penn State University Library's collection.</i>
EX2084	A library catalog MARC record for <i>Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing</i> , available from Penn State University Library at https://mit.primo.exlibrisgroup.com/discovery/fulldisplay?context=L&vid=01MIT_INST:MIT&search_scope=all&tab=all&lang=en&docid=alma9935203535606761

Exhibit No.	Description
EX2085	Wong, Alexander, et al. "Yolo nano: A highly compact you only look once convolutional neural network for object detection. 2019," <i>Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing</i> , available at https://ieeexplore.ieee.org/document/9463524
EX2086	Apte, Maneesh, Simar Mangat, and Priyanka Sekhar. "YOLO Net on iOS." CS231n. Stanford University (2017), <i>available at</i> https://web.archive.org/web/20171118092245/https://cs231n.stanford.edu/reports/2017/pdfs/135.pdf
EX2087	A library catalog MARC record for <i>Procedia Computer Science</i> , <i>available from Penn State University Library at</i> https://catalog.libraries.psu.edu/catalog/51678444
EX2088	Jain, Vaibhav, and Dinesh Patel. "A GPU based implementation of robust face detection system." <i>Procedia Computer Science</i> 87 (2016): 156-163, <i>as sourced from the Penn State University Library's collection.</i>
EX2089	A library catalog MARC record for <i>Proceedings of the 2017 IEEE International Conference on Computer Vision</i> , <i>available from Penn State University Library at</i> https://catalog.libraries.psu.edu/catalog/22615741
EX2090	Lu, Yongyi, Cewu Lu, and Chi-Keung Tang. "Online video object detection using association LSTM." <i>Proceedings of the IEEE international conference on computer vision</i> (2017), <i>as sourced from the Penn State University Library's collection.</i>

Exhibit No.	Description
EX2091	A library catalog MARC record for <i>Proceedings of 2015 IEEE International Conference on Image Processing</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/27695350
EX2092	Apaza-Agüero, Karl, Luciano Silva, and O. R. Bellon. "Mesh segmentation with connecting parts for 3D object prototyping." 2015 IEEE International Conference on Image Processing (ICIP) (2015), <i>as sourced from the Penn State University Library's collection.</i>
EX2093	A library catalog MARC record for <i>Computer Graphics Forum</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/51545295
EX2094	Watanabe, Kouki, and Alexander G. Belyaev. "Detection of salient curvature features on polygonal surfaces." <i>Computer Graphics Forum</i> . Vol. 20. No. 3. Oxford, UK and Boston, USA: Blackwell Publishers Ltd (2001), <i>as sourced from the Penn State University Library's collection.</i>
EX2095	A library catalog MARC record for <i>Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition</i> , available from Penn State University Library at https://catalog.libraries.psu.edu/catalog/39331403
EX2096	Dalal, Navneet, and Bill Triggs. "Histograms of oriented gradients for human detection." 2005 IEEE computer society conference on computer vision and pattern recognition

Exhibit No.	Description
	(CVPR'05). Vol. 1. IEEE (2005), <i>as sourced from the Penn State University Library's collection.</i>
EX2097	Nvidia, GeForce GTX Titan X Specifications, <i>available at</i> https://web.archive.org/web/20200809205411/https://www.nvidia.com/en-us/geforce/graphics-cards/geforce-gtx-titan-x/specifications/
EX2098	Nvidia, CUDA, <i>available at</i> https://web.archive.org/web/20200809115603/https://www.nvidia.com/en-us/geforce/technologies/cuda/
EX2099	A library catalog MARC record for <i>Foundations of Machine Learning</i> , <i>available from Penn State University Library at</i> https://catalog.libraries.psu.edu/catalog/9618647
EX2100	Selections from Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar. <i>Foundations of machine learning</i> . MIT press (2018).
EX2101	Redmon, Joseph, et al. "You Only Look Once: Unified, Real-Time Object Detection", <i>available at</i> https://web.archive.org/web/20150922103408/http://pjreddie.com/media/files/papers/yolo.pdf
EX2102	Iandola, Forrest N., et al. "SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <0.5MB Model Size", <i>available at</i> https://web.archive.org/web/20170128054438/https://arxiv.org/pdf/1602.07360v4.pdf

Exhibit No.	Description
EX2103	Shi, Xingjian, et al. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.”
EX2104	Hu, Wen-Shuai, et al. “Spatial-Spectral Feature Extraction via Deep ConvLSTM Neural Networks for Hyperspectral Image Classification.” IEEE Transactions on Geoscience and Remote Sensing. Volume 58, Issue 6 (June 2020)
EX2105	Shi, Xingjian, et al. “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting.”, <i>available at</i> https://web.archive.org/web/20200915014241/https://arxiv.org/pdf/1506.04214v2.pdf .
EX2106	Hu, Wen-Shuai, et al. “Spatial-Spectral Feature Extraction via Deep ConvLSTM Neural Networks for Hyperspectral Image Classification.”, <i>available at</i> https://web.archive.org/web/20220508083912/https://arxiv.org/pdf/1905.03577.pdf https://web.archive.org/web/20220508083912/https://arxiv.org/pdf/1905.03577.pdf .
EX2107	U.S. Patent App. Ser. No. 62/261,260

I, Nikolaos Papanikolopoulos, of Minneapolis, Minnesota, declare that:

I. INTRODUCTION

1. I have been retained by Patent Owner Airspace Systems, Inc. in this *inter partes* review (IPR) proceeding number IPR2025-00629, filed by Petitioners Axon Enterprise, Inc., Dedrone Holdings, Inc., and Skydio, Inc. I understand that this IPR proceeding relates to U.S. Patent No. 10,514,711 (the '711 patent).

2. I was asked to provide my objective, independent analysis of the '711 patent in view of the prior art references cited in the Petition and to provide my opinion regarding the allegations in the Petition and substitute claims proposed by Patent Owner. To this end, I provided a declaration (EX2025) (the "Initial Declaration"), setting forth my opinions. I incorporate the Initial Declaration herein, by reference, in its entirety.

3. I was informed and understand that the Petitioners objected to certain exhibits cited in the Initial Declaration, as allegedly lacking authenticity and/or foundation. I was asked to respond to these allegations, and accordingly, I provided a Supplemental Declaration (EX2027). I incorporate the Supplemental Declaration herein, by reference, in its entirety.

4. I am further asked to provide my objective, independent analysis of the '711 patent and the proposed substitute claims in view of Petitioners' Opposition to Patent Owner's Motion to Amend and the prior art references cited

in the Opposition, and to provide my opinions regarding the allegations in the Opposition. My response is set forth herein in this Second Declaration.

5. I have based my opinions on the information available to me as of the date of this declaration. I understand that additional information may become available after this declaration is submitted. If I become aware of additional information following my submission of this declaration I may, if I am permitted, offer additional opinions (including by amending and/or supplementing this declaration) taking into account such new information.

6. I am not currently, nor have I ever been, an employee of Airspace Systems, Inc., its predecessors, subsidiaries, parent organizations, or affiliates. I received no compensation for this declaration beyond my normal hourly compensation based on my time actually spent working on this IPR. I will not receive any added compensation based on the outcome of this or any other IPR, or any other proceeding, involving the '711 patent.

II. COMPENSATION

7. I am being compensated as an expert witness in this case at my standard hourly rate in addition to out-of-pocket expenses. I have received no additional compensation for my work on this matter and my compensation does not depend, and has not ever depended in any way, on my opinion as expressed in this declaration and in any testimony that I may give, or on the outcome of this

matter. I have no financial stake in Airspace Systems, Inc., its predecessors, subsidiaries, parent organizations, or affiliates, the ‘711 patent, or the outcome of this IPR or any other matter involving the ‘711 patent.

III. MATERIALS CONSIDERED

8. In formulating my opinions, I have relied upon my training, knowledge, and experience that are relevant to the ‘711 patent. I have also considered the ‘711 patent, patent applications identified in the priority claim of the ‘711 patent, and its file history. In addition, I have reviewed the Petition (“Pet.”) and attached exhibits, including the first declaration by Dr. Singhose and the prior art cited by Petitioners. Moreover, I have considered the Patent Owner Preliminary Response (POPR) and attached exhibits, the Board’s Institution Decision, Patent Owner Response (POR) and attached exhibits, Patent Owner’s Motion to Amend (MTA), and substitute claims that were initially proposed by Patent Owner in the MTA.

9. Additionally, I have reviewed Petitioners’ Opposition to the MTA, the second declaration by Dr. Singhose, and the new prior art cited by Petitioners. I also reviewed transcripts of my own deposition and of Dr. Singhose’s deposition, and the Preliminary Guidance (“PG”) that the Board provided.

10. Furthermore, I have reviewed Patent Owner’s Revised Motion to Amend (RMTA), substitute claims that were proposed by Patent Owner in the

RMTA, Petitioners' Opposition to the RMTA, and the third declaration by Dr. Singhose (EX1039).

11. For the sake of convenience and quick reference, I have reproduced above updated Patent Owner's Exhibit List, listing some of the materials that I considered. I am informed by Counsel and understand that this Declaration will support a Reply that Patent Owner will file. I am further informed and understand that the Reply is also supported by the declaration of June Munford, which demonstrates that the new exhibits submitted with Patent Owner's Reply are authentic and were publicly available as of the dates identified in Patent Owner's Exhibit List. EX2037.

IV. LEGAL PRINCIPLES

12. I am not an attorney. I have been asked to provide opinions on patentability of the claims challenged by Petitioners and substitute claims proposed by Patent Owner. To this end, I relied on certain legal principles that were provided to me by counsel and that are summarized below.

A. Enablement and Written Description

13. I am informed by Counsel and understand that 35 U.S.C. §112(a), requires the patent disclosure as filed to be full and clear so as to enable a POSITA to make and use the invention. I am further informed and understand that the Federal Circuit has interpreted the statutory requirement to mean that a POSITA

should be able to make and use the invention without “undue experimentation.” *In re Wands*, 858 F.2d 731, 737 (Fed. Cir. 1988). The Court stated that a “patent need not disclose what is well known in the art,” and that some experimentation may be necessary. *Id.* at 735, 737. I am informed and understand that what is “undue” may be determined via an eight-factor analysis. *Id.* at 737.

14. Moreover, I am informed and understand that the specification reasonably conveys to a POSITA “that the inventor had possession of the claimed subject matter.” *Ariad Pharmaceuticals, Inc. v. Eli Lilly and Co.*, 598 F. 3d 1336, 1351 (Fed. Cir. 2010). To show possession, “the specification must describe an invention understandable to that skilled artisan and show that the inventor actually invented the invention claimed.” *Id.*

15. Additionally, I am informed by Counsel and understand that 35 U.S.C. §112(b), requires a patentee to present claim(s) that particularly point out and distinctly claim the subject matter which the inventor(s) regard as the invention. I am informed and understand that the Supreme Court has interpreted this requirement to mean that a claim, “read in light of the specification ... and the prosecution history,” must “inform, with reasonable certainty, those skilled in the art about the scope of the invention.” *Nautilus, Inc. v. Biosig Instruments, Inc.*, 572 US 898 (2014).

B. Enablement of Prior Art

16. I am informed by counsel and understand that to “render a claim obvious, the prior art, taken as a whole, must enable a skilled artisan to make and use the claimed invention.” *Raytheon Technologies v. General Electric Co.*, 993 F.3d 1374, 1380 (Fed. Cir. 2021). A “prior art reference asserted under § 103 does not necessarily have to enable its own disclosure, i.e., be ‘self-enabling,’ to be relevant to the obviousness inquiry.” *Id.* “But even though a non-enabling reference can play a role in an obviousness analysis, the evidence of record must still establish that a skilled artisan could have made the claimed invention.” *Id.* at 1381.

17. I am also informed and understand that “[e]nablement requires that the prior art reference must teach one of ordinary skill in the art to make or carry out the claimed invention **without undue experimentation.**” *Elan Pharmaceuticals v. Mayo Foundation*, 346 F.3d 1051, 1054 (Fed. Cir. 2003) (internal quotation marks omitted). “[W]hat constitutes undue experimentation” regarding the disclosure of a prior art reference can be determined using the eight factors described in *In re Wands*.” *Id.* at 1054-55 (citing *In re Wands*, 858 F.2d 731 (Fed. Cir. 1988)).

18. I am further informed and understand that in *Raytheon*, it was an error to “focus[] only on whether [a skilled artisan] is provided with sufficient

parameters in [a prior art reference] to determine, without undue experimentation,” the claimed limitation, “rather than determining whether [that prior art reference] enabled a skilled artisan to make and use the claimed invention.” *Raytheon*, 993 F.3d at 1380. A prior art reference’s “self-enablement (or lack thereof) is not only relevant to the enablement analysis, ... it is dispositive” when “no [] other evidence [is] presented” “to establish that a skilled artisan could have made the claimed” invention. *Id.* at 1382.

V. SUMMARY OF MY OPINIONS

19. Petitioners have challenged independent substitute claims (“SCs” collectively or “SC” individually) 21, 36, and 37 on various grounds. SCs 21 and 36 each requires a machine learning model or a neural network to be executed on-board a drone. A significant flaw in Petitioners’ arguments regarding SCs 21 and 36 is that none of the cited references teaches or suggests this limitation. This is not surprising because in 2017 it was well understood and reported that the implementation of many ML models and neural networks on-board a drone, in particular, was a computationally and memory access intensive task, that was difficult to implement on a conventional, generic processor, typically used by a drone’s on-board system, e.g., to function as a drone controller.

20. Additionally, substitute claim 21 requires a machine learning model running on-board a UAV to provide a bounding box surrounding the detected

features of a target. No prior art teaches or suggests this limitation. Specifically, some prior art references teach obtaining bounding boxes using a neural network (e.g., Pierce, Kumar), but they do not mention the detected features of a target. To the extent one reference (Khan) teaches detecting features of a target, it does so using conventional signal processing techniques and not via machine learning or neural networks.

21. Proposed substitute claim 36 requires the use of two neural networks. Specifically, a first neural network having a high compute cost / robustness and slower speed, for target detection, and a second neural network having a relatively smaller cost and less robustness, but greater speed, for target tracking.

22. The only reference (Ning) upon which Petitioners rely as allegedly teaching this limitation, does not teach or suggest, however, a second neural network that is less costlier, i.e., faster, than the first neural network. This results from a specific configuration of two neural networks in Ning's system, where a YOLO used for object detection provides the essential inputs to an LSTM used for object tracking. Due to this dependence, the LSTM cannot commence its operation until the YOLO provided the needed inputs. The latency, i.e., cost of the LSTM is thus cumulative of that of the YOLO. This makes the LSTM's cost greater than the YOLO's – which is the opposite of what the proposed substitute claim 36 requires.

23. This problem persists regardless of Ning's specific configuration,. A YOLO is typically faster than an LSTM, i.e., the YOLO's cost is generally less than the LSTM's, which is the opposite of what the proposed substitute claim 36 requires. Moreover, in Ning, because the neural networks used for detection and tracking are substantially different from one another, both structurally and functionality-wise, their roles cannot be swapped to address the major high cost / low cost neural networks difference that exists between Ning's alleged neural networks and the neural networks of the proposed substitute claim 36.

24. Substitute claim 37 also requires the use of two neural networks, where the decision of which neural network to use is based on a confidence level. For SC 37, Petitioners do not argue invalidity on the basis of any prior art. Petitioners only argue that SC 37 is not supported by the specification, is not enabled, and is indefinite. These arguments have a common flaw. In making them, Petitioners have interpreted SC 37 as requiring the decision of which neural network to use for processing a current image to be based on the processing of the current image captured by the UAV. This interpretation is erroneous; per the proposed substitute claim 37, this decision is based on a confidence level obtained from the processing of an immediately preceding image.¹

¹ The decision may also be based on the confidence level(s) obtained from the processing of one or more images preceding the immediately preceding image.

25. The asserted grounds for independent substitute claims 21, 36, and 37 fail for additional reasons as well, as I discuss below. Because these claims are valid, their dependent claims are also valid.

26. The features of SC 21 enable drone-to-drone detection and tracking because they employ a refined technique for determining a three-dimensional distance to a moving target, such as a drone. In particular, a bounding-box-based estimation is improved by taking into account the physical size based on a drone model type that is determined using an on-board neural network. Moreover, the features of SCs 36 and 37 enable drone-to-drone detection and tracking because they facilitate balancing detection quality (robustness) and latency (computation cost). A slower neural network may be used intermittently to ensure quality, while a faster (high refresh rate) neural network may be used to maintain an effective lock on a fast-moving target. UAVs and UAV operation according to these SCs can tackle innovatively a highly nonlinear three-dimensional problem in vision-based control.

VI. THE '711 PATENT – INNOVATIVE ASPECTS OF PROPOSED SUBSTITUTE CLAIMS 21, 36, AND 27, AND CLAIM 12

A. Overview

27. I discussed the innovative aspects of the '711 in my Initial Declaration. EX2025 ¶¶65-83. Upon reviewing Petitioner's Reply (Rep.) to the Patent Owner Response (POR), Petitioner's Opposition (Opp.) to Patent Owner's

Motion to Amend (MTA), and the Second Declaration by Dr. Singhose (EX1030), it appears to me that the Petitioners have misunderstood the key innovative aspects disclosed and claimed in the '711 patent. I therefore describe these key aspects in the context of claim 12 and the proposed independent substitute claims (hereinafter, "SCs" or "SC," individually) 21, 36, and 37.

28. In 2017, Patent Owner ("PO") entered into a contract with the United States Department of Defense to develop a fully autonomous interceptor drone. *See* EX1013, ¶¶9, 10. Up until this time, drone flights were usually controlled from a ground station, e.g., by a human operator, software, etc. *See* EX1001, 1:27:31. A main objective of PO's development effort was for the interceptor drone to "**complete** a desired flight operation" even "if **communication** between the aerial vehicle and the flight controller **is lost or becomes unreliable**." *Id.* 1:34-37. PO achieved this objective, and in the process earned recognition from the Smithsonian Institute, by solving three challenges that all arose at a "near" stage of a larger autonomous navigation process.

29. The "near" stage commences when the interceptor drone is finally near the target, has stopped its autonomous navigation based on the target location received from the ground system, and has started engaging in autonomous navigation using its image based navigation system. The '711 patent describes that system as:

When an unauthorized aerial vehicle is detected (e.g., using sensors of ground station 104 and/or ground-based sensor 106), interceptor aerial vehicle 102 may be deployed to interdict/capture the unauthorized aerial vehicle. An example of the unauthorized aerial vehicle is target aerial vehicle 110 (e.g., a drone, a multicopter aircraft, an airplane, a UAV, a helicopter, or any other vehicle capable of flight). The detected location of target aerial vehicle 110 may be provided to interceptor aerial vehicle 102 to allow it to automatically and *autonomously fly* towards the direction of the detected location. *However, once interceptor aerial vehicle 102 is near target aerial vehicle 110, the location data from ground-based sensors may not be accurate or updated fast enough* for interceptor aerial vehicle 102 to get close enough to the moving target aerial vehicle 110 to reliably deploy a capture mechanism.

Id., 4:7-22.

30. At this “near” stage, the challenge was how to enable an interceptor to autonomously navigate toward the target, which meant not being reliant **in any way** on the ground station for the information required for such autonomous navigation, whether for control, guidance, etc. It was known for a ground-based system to apply machine learning to an image captured by a drone to define a bounding box outlining an area within the image that includes the image portion of the target, and to then use the bounding box size and location within the image to calculate a relative distance to the target. It was also known that in this context, the

accuracy of the calculation of the relative distance to the target based on a bounding box can be increased by accounting for the target's actual physical size.

31. No one knew however, how to perform these computations on an interceptor drone. Prior systems typically either: (i) employed traditional signal processing on-board the interceptor drone, which was problematic because in the high-noise environment of a drone, accuracy of on-board signal processing can be inadequate, or (ii) performed machine learning / neural network based computations at a ground station, which then provided the results of the computation to the drone.

32. In contrast, the '711 patent teaches a machine learning model / neural network on-board the interceptor drone with the following characteristics:

[T]hus the size of the detected target within the image can be used as a proxy for the distance to the target. However, different types of target aerial vehicles may be different sizes and the actual physical size of the target needs to be taken into account when using the target size within the image as a proxy for relative distance. In some embodiments, the **machine learning model has been trained using different examples of types of aerial vehicles** (e.g., different models, types, manufacturers, sizes, etc.) to be able to classify and identify the specific type of the detected target aerial vehicle and a corresponding size profile corresponding to the specific type.

Id. 17:45-58; *see id.* 17:8-19:3 (describing location determination).

33. PO's interceptor drone was thus the first to use its own on-board machine learning system to determine a size profile of the target when relying on image-based navigation, which in turn enabled it to be the first drone to autonomously navigate toward a target without needing any control, guidance, or data / information (e.g., the size of a detected target drone) from the ground station when navigating in the "near" stage of autonomous navigation.

34. PO also had to enable autonomous navigation at the "near" stage to perform all the required machine-learning computation needed to perform machine learning without help from a ground system. By severing the interceptor drone's dependence on the ground system, PO put itself in uncharted territory in that its interceptor – a drone – was now saddled with an unprecedented computational burden. It was well-known at the time, and it is still true today, that machine learning systems, and neural networks in particular, require a substantial computation power, special purpose processors such as GPUs and/or specialized software frameworks, which can be readily provided in a server at a ground station but not on more limited computational platforms such as those available on a drone.

35. PO's interceptor drone had to address a wide range of computational challenges that arose especially when performing "near" stage autonomous navigation, i.e., without any help from a ground station. The interceptor drone had

to meet this challenge on a hardware platform that had resource constrained by power, weight, size, and heat dissipation constraints. That limited the number of processors and the amount of memory that can be provided on-board a drone. Computation speed was paramount, however. In spite of having limited resources, the computations needed to be extremely fast and accurate; otherwise, the autonomous navigation would fail.

36. Specifically, two interceptor-drone specific and very different computational challenges had to be solved. The first was detecting a target in an image for the first time (or just after the interceptor drone had lost its lock on a previously detected target). This involved distinguishing a target from a near-infinite range of other objects and from the background objects the interceptor drone would capture with its camera. *See id.*, 17:18-30 (describing using machine learning models to solve this problem, at least in part).

37. The second problem was maintaining a lock on an already-acquired target that is constantly moving, using a camera aboard a drone that was also constantly moving. *See id.*, 3:21-23 (describing a moving target), 4:17-22 (describing that in the “near” stage, location data received from the ground is not accurate enough and updated frequently enough), 17:17-37 (describing target location and distance determination using machine leaning models), 25:66-26:4

(describing a multimodal sensor empowered awareness engine for the recovery of a lost target), 27:4-10 (same), FIG. 5, 22.

38. To address these two different computational challenges on a computation budget, which, in this context, at least meant the time available for processing an image, PO invented a drone operation logic / computing system that had the ability to select between one of two different computation modes **when processing a captured image** (e.g., a “spatio-temporal frame” or “video frame”): (1) a high-quality prediction (high robustness) mode when processing the image for the initial target detection, or for recovery if a detected target is lost, and (2) a high refresh (frame) rate, i.e., low compute cost per frame, mode when processing the images for maintaining lock on a detected target.

39. Specifically, the '711 patent states:

On a fixed compute budget there is a negative relation between robustness of an algorithm and the cost and/or latency of the algorithm per spacio-temporal frame. Tight restrictions on available compute cycles limit the responsiveness of the system, especially on autonomous or semi-autonomous drones. **By combining multiple algorithms in a way that achieves high combined frame rate with higher quality prediction, the system improves outcomes on a limited resource budget.** Deep neural networks (DNN) may be executed to periodically detect objects and distinguish targets. A main high resolution tracking engine executes high-speed

feature-based tracking based on disparity of similarity features with live adaptation.

Id. 21:52-65.

40. The '711 patent further states:

Once an object is **detected in high-resolution mode** (i.e. using DNN or manual input) a tracker algorithm takes control and maintains lock on the target. Applying different tracking algorithms and DNN based detections of a target in the video frame provides robustness at a high compute cost. A **reduced resource background tracker may incrementally predict the location** of a target in the frame with low compute cost and average robustness. This optimization enables the coexistent application of advanced machine vision algorithms in addition [to] specialized lower cost algorithms.

Id. 21:67-22:10.

41. By implementing two different computation modes potentially applicable to any captured image, PO imbued the interceptor with an extensive range of computational capabilities, notwithstanding being subjected to severe power, weight, size, heat dissipation, and other hardware constraints. The interceptor drone thus became computationally capable of autonomously navigating to the target, especially during the “near” stage of autonomous navigation.

42. Having developed an interceptor that by performing the necessary **computations** on-board could itself generate all the **information** it needed to

achieve autonomous navigation (i.e., without help from the ground system and notwithstanding severe hardware constraints), PO once again found itself in uncharted territory – now having to solve a **control** problem that was also not previously contemplated let alone solved in the drone industry. PO now needed to figure out when to use and how to **control** the various computational resources on the interceptor as it engaged in “near” stage autonomous navigation, including how to schedule processing schemes that each ran in a very different computation mode in terms of robustness, e.g., quality of target detection in highly noisy environment versus cost, and in terms of the rate at which new images / frames can be processed.

43. This was a fundamentally new problem complicated by the highly nonlinear dynamics in which an interceptor was operating when engaged in autonomous navigation. For instance, the ’711 patent states: “because the target UAV can move, the interceptor aerial vehicle needs to be able to follow the target UAV and get close enough until the target UAV is within range of a capture system.” *Id.* 3:21-24. The ’711 patent further states: if “the image of the target is lost, the reference system may fall back to the previous[ly] received location-based navigation until the target is detected again in a captured image.” *Id.* 16:31-35.

44. In addition, during image-based autonomous navigation, “the flight path of the interceptor aerial vehicle [may] be dynamically adjusted to track and

follow a moving target, allowing the interceptor aerial vehicle to come within a threshold distance where an interdiction/capture system ... can be effectively deployed.” *Id.* 16:54-61. The ’711 patent also discloses “a multimodal awareness system [that] provides for a fast recovery of lost targets.” *Id.* 25:67-26:6.

45. The interceptor-drone solved the difficult problem of recovering a lost target while autonomously navigating towards the target by sensing, learning, and making a **control decision**, such as scheduling one of several different processing schemes to process an image in real time, on hardware constrained by the size, weight, and power limits of a small drone, and without relying on any control commands from a ground station. Specifically, the interceptor drone implemented a processing control loop in which the interceptor continuously monitored the quality of its computations pertaining to target detection and tracking, and balanced the computational cost, to determine which of several computational modes (defined by robustness versus cost (e.g., in term of processing time) as described above). On-board image processing was performed using the control loop.

46. For example, the control loop is used to apply one of several neural networks implemented on the interceptor drone to an image, again depending on the quality of the outputs that are being monitored in the loop, and the respective costs of generating those outputs using different neural networks. Specifically, in

describing neural networks having different robustness and computation cost / refresh rates, the '711 patent states:

Processing scheme examples for a **group (i) High cost (low refresh rate)—high robustness** can include modified tiny You only look once (YOLO) Convolutional Neural Networks (CNN) on 448 pixels input tiles in the frame; a modified SSD (Single Shot Detection) CNN on 300 pixels multi-scale classification; a modified Faster R-CNN (Region Proposal Networks), segmentation analysis of the frame+ classifier for each candidate, etc.

Processing scheme examples for a group (ii) & (iii) can include using the SqueezeNet superfast rejection CNN as the validation method on the estimated location produced by the algorithms. ...

Group (iii) Low-cost—low robustness processing scheme may have no confidence measure and depend solely on the validation method to determine a performance score and/or detect a failure. Group (iii) Low-cost—low robustness processing scheme examples can include: extrapolation of location of object from past locations (e.g., no confidence measure (always max), depend on validation method to reject estimation, etc.); SqueezeNet super fast rejection CNN; template matching to the last known object appearance based on past detections, etc.

Id. 28:40-29.

47. The '711 patent also describes two processing schemes. In the first one, a low robustness, high refresh rate processing is employed upon initial target

detection until it is lost. For initial detection and loss recovery, a high robustness, low refresh rate processing is employed. Specifically:

Once an object is **detected in high-resolution mode** (i.e. using DNN or manual input) a tracker algorithm takes control and maintains lock on the target. Applying different tracking algorithms and DNN based detections of a target in the video frame provides robustness at a high compute cost. A **reduced resource background tracker may incrementally predict the location** of a target in the frame with low compute cost and average robustness. This optimization enables the coexistent application of advanced machine vision algorithms in addition [to] specialized lower cost algorithms.

Id. 21:67-22:10.

48. In the second processing scheme, the quality of object detection, a confidence level, is obtained in either type of processing – high or low robustness / high or low cost. If the current confidence level is high enough, as determined by a threshold, the processing may switch to the low robustness / cost process. If the confidence level becomes low, high robustness / cost processing is employed.

An example implementation includes method comprising optimizing resources [for] processing a video data stream from a mobile capture device using a set of processing schemes to track one or more items of interest. A **performance score** is associated with each processing **scheme confidence** to track the one or more items of interest. The method includes **repeatedly determining** an active processing scheme based on the processing scheme with the highest performance

score from the set of processing schemes. In response to the **performance score** of the active processing scheme **failing to satisfy a threshold**, the method selects another processing scheme to process the video data stream. Processing the video data stream can include identifying one or more items, classifying each of the items; and tracking one or more of items as an item of interest based on the classification.

Id. 22:18-32. Additionally,

According to the example implementations, if the confidence number is higher than an activation number, an action is performed to stop or change a running group (i) and select or switch to group (ii).

Id. 28:1-5. Such a control loop was wholly absent from interceptor (and other types of) drones before its invention by PO.

49. PO's solutions to the aforementioned ground station reliance, computational-deficit, and control problems – enabled PO to deliver to the U.S. Army a first-in-breed interceptor drone that was recognized by the Smithsonian Institute. PO's solutions to these problems are described in the '711 patent, and proposed substitute claims 21-37 are directed towards various important features of those solutions. I elaborate on the problems identified above and the solutions disclosed in the '711 patent further below.

B. Drone’s On-Board Use of Machine Learning Models / Neural Networks

50. In order to fly in a safe and effective manner, an interceptor drone must be able to detect, by itself and while in flight, targets that may also be airborne and moving rapidly in a three-dimensional space.² Otherwise, the drone must rely on a ground-based system to detect and track targets and to continually provide the target locations to the drone. This can be error-prone, *see* EX1001, 16:5-10 (stating that the location received from a ground station “may not represent the exact location of a moving target”), and at a minimum, requires continuous ground communication.

51. Because the drone and the target may be moving rapidly relative to each other, the on-board target detection needs to be both fast and accurate, and additionally, resilient to a high noise input data, such as radar / lidar images and video and visual images captured by a flying drone. In a high noise environment, ML systems / neural networks often perform more accurately compared to the traditional signal / image processing techniques. *See* EX1025 ¶¶71-72. But, many fast and high accuracy ML systems and neural networks that were available around and before 2017 required server-type computing power and could not be placed on-

² This is true of non-interceptor drones as well because a target can also be an obstacle to be avoided.

board a drone. Relying on a server-based system would require continuously transmitting or streaming the data (images, video) captured by the drone to the ground system, and receiving results of the target detection from the ground system. This too requires continuous ground communication.

1. Challenges in Drone’s On-Board Use of ML Models / Neural Networks

52. Before 2017, the use of a generic CPU / image processor for performing traditional signal / image processing on-board a drone was known. *See, e.g.,* EX1009, 7:27-32. It is tempting to think that because traditional signal / image processing techniques and ML / neural networks can solve the same problem, i.e., object detection and tracking, it would be straightforward to replace the former with the latter. This is far from the truth, however. Before and around 2017, and even much later, it was well understood that such a replacement or substitution is far from straightforward. Before 2017, a practical implementation of neural networks on-board a drone for target detection and tracking was not considered feasible.

53. For example, Vanhoucke notes that the “sheer size of [neural] networks can represent a **challenging computational burden**, even for modern CPUs. For this reason, GPUs are routinely used instead to train and run such networks.” EX2039, Abstract. Vanhoucke does disclose a technique that can significantly improve the performance of a neural network executed on a CPU. But

that improvement requires **two substantial modifications**. First, the CPU must support “SSE” instructions, *id.*, i.e., “Intel® Streaming SIMD Extensions.”

EX2040; EX2041 at 1 (p. 47). Second, the neural network must be implemented using fixed-point arithmetic instead of the typical floating-point arithmetic.

EX2039 at 1 (improvement “by a factor of 3 using fixed point arithmetic and SSSE3 / SSE4 instructions”), 5 (describing fixed-point implementation using linear quantization).

54. The implementation of an ML model / neural network on a generic CPU / image processor was not considered practical, at least in the context of computer vision, for several reasons. First, the number of computations involved in the execution of ML models and neural networks is generally **one or even two magnitudes of order greater** than the number of computations required by conventional signal / image processing techniques. EX2042 at 2 (Table II showing that a traditional image processing technique such as histogram of oriented gradients (HOG), requires about 0.7 giga operations per mega pixels (GOP/Mpixel), but two convolutional neural networks, CNN, require 25.8 and 610.3 GOP/Mpixel, respectively).

55. Due to this significantly greater computational requirement of ML / neural networks, it was also known before 2017 that the implementation of ML models / neural networks usually necessitates highly specialized processors such as

graphics processing units (GPUs), and that execution by a generic CPU / image processor would not be practical. *See* EX1001, 26:16-33, 26:42-44, FIG. 20 (disclosing GPU implementation); EX1036 at 48 (p. 779) (disclosing “Titan X GPU implementation”); EX2039, Abstract (“GPUs are routinely used instead [of CPUs] to train and run [neural] networks”); EX2043 at 552 (machine vision techniques “mostly need vast quantity of GPU power” and “non-GPU computer[]” implementations of “YOLO” and “Faster R-CNN” are “unsuitable for real-time application”).

56. Second, in addition to the substantially greater computational complexity, the execution of a neural network on a generic CPU / image processor that was typically found on-board drones, also requires a suitable framework, such as TensorFlow, Darknet, Darkflow, OpenCV, Keras, Metal, etc. *See* EX2043 at 2 (p. 553); EX2044 at 6-7. A neural network, however, is often designed for one particular framework, and that framework is not readily implementable on a generic CPU or a digital signal processor / image processor. *See* EX2043 at 2 (noting that “DarkFlow [] and OpenCV ... **only** ... are favorable for CPU based YOLO,” which is a neural network); *see* EX1034, 4:48-49; EX1036, Abstract. Even though CPU-favorable frameworks were known, installing such a framework for use by a generic CPU, e.g., using the Windows operating system, can be “really

a complex task” and may not yield satisfactory results. EX2043 at 2 (describing the results from the installation of DarkFlow as “terrible”).

57. Third, even if it were possible for a suitable framework to be provided to a generic CPU / image processor, the parameters of the neural network designed for one framework are not directly applicable to another, which means one would need to perform the very difficult task of converting a neural network’s parameters before deploying it on a new framework suitable for a generic CPU / image processor. For example, Hollemans states:

YOLO is written in Darknet, a custom deep learning framework from YOLO’s author. The downloadable weights are available only in Darknet format. Even though the source code for Darknet is available, **I wasn’t really looking forward to spending a lot of time** figuring out how it works.

58. Thus, the neural network parameters available for a certain framework need to be modified using specialized software and/or math before being able to deploy it to a generic CPU / image processor. *See* EX2044 at 7-10 (describing converting Darknet “weights” for tiny YOLO into Keras weights using an available “YAD2K” script, and then converting Keras weights into Metal weights using a custom script and “a bit of math,” involving specific mathematical formulae).

59. In summary, at least (i) the substantial (one or two orders of magnitude) increase in computational complexity, (ii) the installation of a suitable

framework, and (iii) customized conversion of neural network parameters for use by a suitable framework were **significant obstacles** before 2017 (and they remain as obstacles even today) that made and continue to make replacing / augmenting traditional signal / image processing by a generic CPU or image processor on-board a drone with an ML model or neural network a hard, if not a formidable challenge.

2. '711 Patent's Use of ML Models / Neural Networks On-Board a Drone

60. The inventors of the '711 patent avoided the above-discussed challenges by providing a GPU on-board the interceptor drone, and also by deploying in its interceptor drone ML models such as “support vector machine, soft max classifier, autoencoders, naive bayes, logistic regression, decision trees, random forest.” *See* EX1001, 7:27-32, 26:16-33, 26:42-44, FIG. 20. Regarding neural networks, the '711 patent discloses three kinds of neural networks – described as high cost, medium cost, and low cost, where the “cost” is the computation cost, such that a neural network having a high computation cost can process data at a slower rate, i.e., a low refresh rate. *Id.* 22:3-5, 27:53-64. The examples of neural networks that the '711 patent provides, such as modified tiny YOLO, Faster R-CNN, SqueezeNet super fast rejection CNN, etc., are well-suited for image detection and tracking by a drone. *See* EX1001, 28:40-29:3.

61. Even the “high cost” neural networks that the ’711 patent identifies, such as tiny YOLO and Faster R-CNN, have significantly lower computational complexity than their full-fledged counterparts, such as YOLO and Fast R-CNN, and can therefore be executed on-board a drone. For example, Fang discloses that YOLO-v1 and v2 require over 40 and 34 billion floating point operations per second (BFLOP/s), respectively, while Tiny-YOLO-v2 and v3 require less than 7 BFLOP/s. EX2045 at 8 (p. 1942). Ren discloses that the Faster R-CNN has a different structure and execution speed compared to Fast R-CNN. *See* EX2046 at 1 (describing an “algorithmic change” to Fast R-CNN, featuring “computing proposals with a deep net” by “shar[ing] convolutional layers with state-of-the-art object detection networks”), 7 (describing speed-up from over 1-2 seconds to 320 milliseconds).

62. Furthermore, the ’711 patent also describes using and switching between two different kinds of neural networks, discussed above as part of a **control** loop. *See supra* § VI.A. One manner of switching is based on the phase of the operation. When no target is detected, a high robustness / high cost neural network (which is nevertheless significantly smaller and faster than its full-fledged version, and can be executed on a drone) is used. Once a target is detected, it is tracked using a lower robustness / lower cost neural network that has a higher refresh rate. This scheme can ensure that targets are detected initially, and

thereafter, image streams are analyzed at a high rate, minimizing the likelihood of losing a detected target that may be moving rapidly in a three-dimensional space.

63. Another manner of switching is based on the confidence in the accuracy of the operation. If target detection / tracking is being performed by a high robustness, low refresh rate neural network, and if the confidence score associated with the detection of the target is high, for subsequent target detection / tracking, the drone switches to using a neural networking having a lower robustness but higher refresh rate. If the confidence score associated with the detection by the low robustness neural network becomes low, however, for subsequent target detection / tracking the drone switches to using a high robustness rate neural network. Such switching increases the likelihood of detecting targets in the vast space surrounding a drone, and minimizes the likelihood of losing the detected target.

64. SCs 21, 36, and 37 are all directed to using a ML model / neural network on-board a drone, so that the need for ground communication is minimized, which minimizes the risk of mission failure when the ground communication is unreliable or unavailable. SCs 36 and 37 are directed to on-board use of two neural networks, with SC 36 being directed to phase-based switching between two neural networks and SC 37 being directed to confidence-score based switching.

VII. SPECIFICATION OF THE '711 PATENT FULLY SUPPORTS ALL PROPOSED SUBSTITUTE CLAIMS

A. Specification Fully Supports Proposed Substitute Claim 37

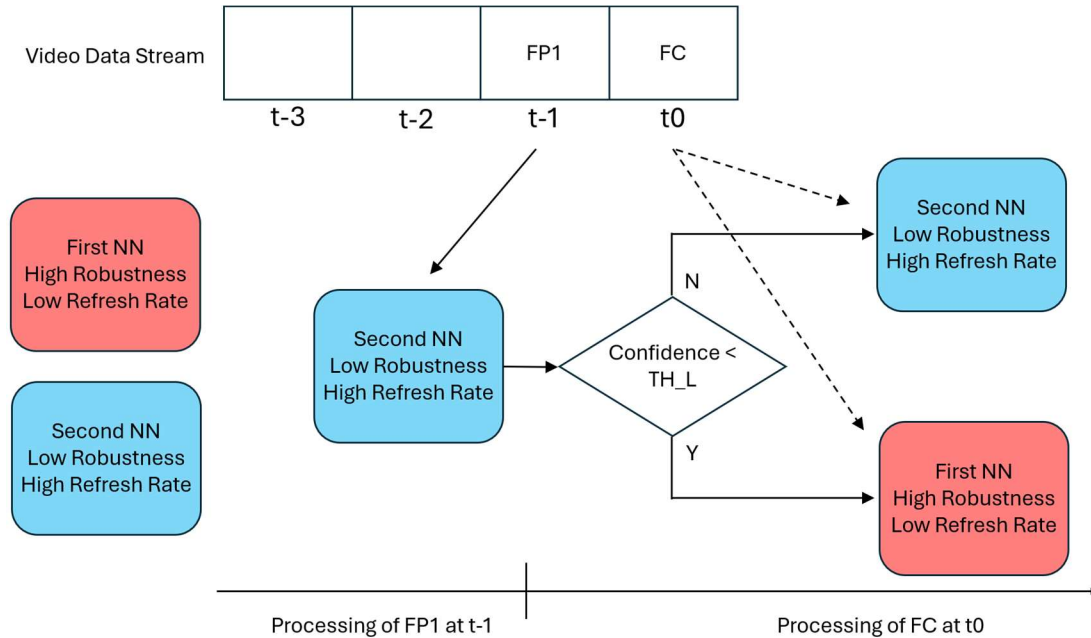
1. Specification Discloses Image-by-Image Selection of a Neural Network

65. Petitioners contend that SC 37 requires “image-by-image decision-making based on [both] the immediately preceding image and the current image,” and allege that the specification does not support such a processing scheme. Opp. at 1; EX1039, ¶76. Petitioner’s contention is partially but not entirely correct, and the allegation is false.

66. As a threshold matter, claim element [37.e1] recites “making a selection, from between the first and the second neural networks, of only one of the first and the second neural networks, based on a confidence level *for* a detection of the target in the current image.” Thus, the decision-making process, i.e., the selection of only one neural network, either the first one or the second one, is made based on a confidence level that is determined “*for* a detection of a target in the current image.” The confidence level is not *based on* the detection of the target in the current image. Rather, the confidence level obtained from the detection of a target in the immediately preceding image informs the confidence *for* a detection in the current image, if the same processing scheme that were used to process the immediately preceding image were also used to process the current image. As

such, the decision making process is not based on “the current image.” Rather, it is based on the “immediately preceding image.”

67. I further explain the selection process below with reference claim element [37.e3] and Demonstrative 1 below.



Demonstrative 1

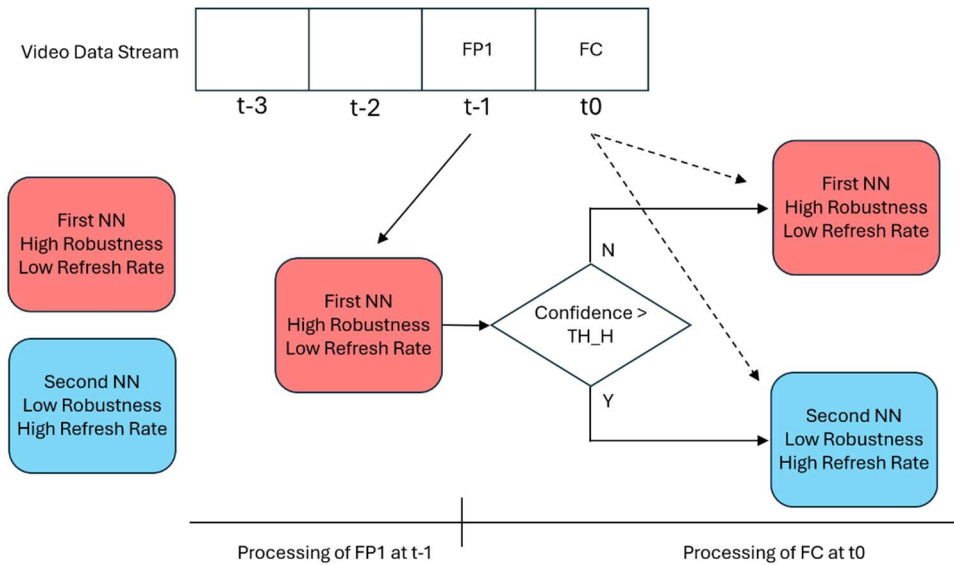
68. The video data stream includes several video frames. A frame labeled FC is the current frame or image that was captured at time t0; the immediately preceding frame / image, that was captured at time t-1, is labeled FP1. Two NNs, respectively labeled “First NN” (marked red) and “Second NN” (marked blue), are available for processing any of the frames of the video data stream. First NN is a high-robustness, low refresh-rate NN, and Second NN is a low-robustness, high refresh rate NN. As claim element [37.e3] recites, suppose during the processing of

FP1, “Second NN” were used (“*when the second neural network was selected and applied to the immediately preceding image*”).³ Upon processing FP1 and attempting to detect a target therein, a confidence level is obtained based on that processing.

69. Before processing of the current frame FC commences, the confidence level is compared to a low threshold (TH_L) of a threshold range. If the confidence level is below the low threshold, it is determined that continuing with Second NN may not yield effective object detection. As such, First NN, having a high robustness, is selected for processing FC (“*(i) selecting and applying the first neural network to the current image when the confidence level is below a first threshold*”). Otherwise, the use of Second NN would continue, where, like FP1, FC would also be processed by Second NN (“*and otherwise (ii) selecting and applying the second neural network to the current image*”).

70. Next, I discuss claim element [37.e4] referring to Demonstrative 2 below.

³ Italicized language in parenthesis is the relevant language of claim element [37.e3].



Demonstrative 2

71. As claim element [37.e4] recites, suppose during the processing of FP1, “First NN” were used (“*when the first neural network was selected and applied to the immediately preceding image*”). Upon processing FP1 and attempting to detect a target therein, a confidence level is obtained based on that processing.

72. Before processing of the current frame FC commences, the confidence level is compared with to a high threshold (TH_H) of the threshold range. If the confidence level is above the high threshold, it is determined that switching to Second NN may yield performance improvement (e.g., quicker processing of the frames of the video data stream while still achieving effective object detection). As such, Second NN, having a low robustness but high refresh rate, is selected for processing FC (“*(i) selecting and applying the second neural network to the*”).

current image when the confidence level is above the second threshold”).

Otherwise, the use of First NN would continue, where, like FP1, FC would also be processed by First NN (“*otherwise (ii) selecting and applying the first neural network to the current image*”).

73. The discussion above of claim elements [37.e3] and [37.e4] shows that SC 37 does require an image-by-image decision making, but the decision is not based at all on the current image, as Petitioners contend. Rather, the confidence level that is used for decision making is obtained from the processing of the immediately preceding frame / image. The specification discloses the processing scheme that is described above and recited in claim elements [37.e3] and [37.e4].

74. In particular, the specification states: “[o]n a fixed compute budget there is a negative relation between robustness of an algorithm and the cost and/or latency of the algorithm per spacio-temporal frame.” EX2019, ¶109. A spacio-temporal frame in image processing is a frame, i.e., an image, that is identified by a spatial component and a temporal component. The spatial component identifies a camera in a multi-camera system. *See id.* ¶108 (the “spacio-temporal awareness engine utilizes a multi-camera view”). The temporal component identifies the time or a sequence number in a series of frames captured by the camera identified by the spatial component. In Demonstratives 1 and 2, for the sake of simplicity, it is

assumed that only one video data stream is captured using a single camera, where the frames that are depicted were captured at times t-3, t-2, t-1, and t0, respectively.

75. As the specification describes, the “negative relation between robustness of an algorithm and the cost and/or latency of the algorithm” exists “*per spacio-temporal frame*.” EX2019, ¶109. In this context, a negative relation means that as the robustness of the algorithm increases, its processing time / latency *per frame* also increases, i.e., the rate at which the frames in the video data stream can be processed decreases. Thus, the specification discloses that a trade-off between robustness and processing speed exists on a per-frame, i.e., per image basis.

76. Additionally, the specification states:

An example aspect includes *optimized* video processing scheme *scheduling*. In an example implementation, processing schemes include localization algorithms⁴ that *process video frames* of one or more video data streams. The *processing schemes produce* a prediction of objects *within the video frame*, as well as a *confidence number* that is a measure of quality for that prediction.

Id. ¶141. The specification also states that “[b]y *combining* multiple algorithms in a way that achieves high *combined* frame rate with higher quality prediction, the system improves outcomes on a limited resource budget.” *Id.* ¶109.

⁴ In computer vision, localization means determining where, if at all, an object of interest is located within a captured frame / image.

77. Thus, optimizing the scheduling of the video processing scheme means determining a manner in which different algorithms are used for the processing of a video data stream. Because the trade-off between robustness and processing speed exists on a per spacio-temporal frame basis, and a “high *combined* frame rate” (i.e., an aggregate frame rate of processing two or more different algorithms) is achieved, the specification teaches that the optimized schedule identifies the most suitable algorithm on a “per spacio-temporal frame” basis i.e., for each frame / image in a video data stream to be processed.

78. In particular, the specification states the “processing schemes produce a prediction of objects *within the video f[r]ame*, as well as a confidence number that is a measure of quality for *that prediction*.” *Id.* ¶141. The ’711 patent’s teaching of prediction (e.g., detection of a target) “*within the video f[r]ame*” and of providing the corresponding confidence level for each frame inform that in *each frame* / image, a target is detected and a corresponding confidence number is obtained, which is then used to select the most suitable processing scheme, i.e., a neural network, for the next frame / image to be processed.

79. The fact that the neural network selection and processing is performed on an image-by-image basis is further supported by the specification. For instance, the specification discloses “modified tiny You only look once (YOLO)” and “SqueezeNet super fast rejection CNN” as examples of a neural networks that may

be used per SC 37. *See* EX2019, ¶¶150-151; RMTA at 23. Regarding the use of YOLO, the specification states that “You only look once (YOLO) Convolutional Neural Network[] (CNN) [operates] on 448 pixels input tiles *in the frame*.”

EX2019, ¶¶150; *see* REDMON-YOLO-2015 at 1 (Figure 1 (“[o]ur system (1) resizes the input *image*”); EX2062 at 6 (describing the first layer as “input image”). The specification also discloses “DNN based detections of a target in the *video frame* provides robustness at a high compute cost.” *Id.* ¶110.

80. Thus, the specification describes neural networks that as in SC 37 use and process each single frame, i.e., each image. Accordingly, as explained above with reference to claim element [37.e1], *see supra* ¶66, the specification discloses image-by-image decision-making that is based on the “immediately preceding image,” but not in any way on the “current image.” Because the system operates on a video stream and repeatedly evaluates confidence per frame, the confidence used to decide a switching between processing schemes (e.g., neural networks) necessarily comes from the immediately preceding frame (and in some cases additional preceding frame(s)) when operating on *per spacio-temporal frame basis*. *See* EX1001, 27:26-32 (“processing schemes produce a prediction of objects *within* the video frame, as well as a confidence number”); *see id.* 21:52-54 (“On a fixed compute budget there is a negative relation between robustness of an algorithm and the cost and/or latency of the algorithm *per spacio-temporal frame*.”).

81. Petitioners’ also argue that the specification discloses “processing video stream ‘frames,’ “examining processing schemes ‘from group to group,” and switching between groups.” Opp. at 1; EX1039, ¶77. To the extent Petitioners interpreted this disclosure as teaching processing several frames or groups of frames at a time by a selected neural network, such an interpretation is flawed.

82. The specification organizes the different processing schemes into three groups and the process of selecting a processing scheme / neural network “cycles through the processing schemes.” EX2019, ¶¶145-146. Upon switching to a particular group based on the confidence level obtained from the processing of the immediately preceding frame, one processing scheme is selected from the chosen group, and the current image is processed by the selected process. *Id.* ¶146 (a “[h]ight cost ... algorithm can select a cheaper processing scheme *from* the group of (ii) [m]edium cost ... algorithms”). Thus, the groups that the specification describes are not groups of frames / images, and do not describe what is processed, which is one frame at a time, as discussed above. The groups are collection of processing schemes, one of which is selected to process the current frame. *See id.* ¶¶150-152.

83. Dr. Singhose argues that “there is no teaching of how to select the neural network applied to the current image as a function of whether the first neural network or the second neural network was applied to the immediately

preceding image.” EX1039, ¶77. This argument has no basis because SC 37 does not recite such a limitation. As claim elements [37.e3] and [37.e4] recite, the only conditions upon which a neural network is selected for the current image are: “when the confidence level is below a first threshold” and “when the confidence level is above the second threshold.” The selection of a neural network is not a function of whether the first neural network or the second neural network was applied to the immediately preceding image.

84. In fact, the language of claim elements [37.e3] and [37.e4] makes it clear that the neural network selected for processing the current image does not depend on the neural network that was used to process the immediately preceding image. Specifically, claim elements [37.e3] and [37.e4] recite:

Claim Element [37.e3]	Claim Element [37.e4]
when the second neural network was selected and applied to the immediately preceding image, making the selection comprises:	when the first neural network was selected and applied to the immediately preceding image, making the selection comprises:
(i) selecting and applying the first neural network to the current image when the confidence level is below a first threshold ..., and otherwise	(i) selecting and applying the second neural network to the current image when the confidence level is above the second threshold, and otherwise
(ii) selecting and applying the second neural network to the current image	(ii) selecting and applying the first neural network to the current image

85. The side-by-side comparison above shows that regardless of whether the first or second neural network was applied to the immediately preceding image, the selection of the neural network for the current image depends only on the respective confidence level based conditions. If the applied condition (in the first row in the table above) fails, whichever neural network was selected for the immediately preceding image remains selected for the current image.

86. The specification also shows that the selection of the neural network for the current frame / image depends not on which neural network was selected for the immediately previous frame / image, but on whether a confidence level is below a low or above a high threshold. The specification states:

If the confidence is below low threshold, falls outside of a range, ... group (i) processing is performed. If the confidence is determined to be high, then group (iii) algorithms can be applied to optimize resources.

EX2019, ¶147. Thus, contrary to Dr. Singhose's (or Petitioners') argument the neural network applied to the current image is not a function of whether the first neural network or the second neural network was applied to the immediately preceding image. Rather, it is a function of the confidence level and the applicable threshold.

87. In summary, the specification fully discloses image-by-image decision-making, i.e., selection of a neural network, that is based on the “immediately preceding image” as claimed.

2. Specification Discloses Thresholds and Ranges

88. Claim element [37.e3] recites “*a first threshold of a range defined by the first threshold and a second threshold greater than or equal to the first threshold.*” Petitioners allege that the specification does not support this limitation. Opp. at 1-2; EX1039, ¶79. Petitioners also allege that per the claim limitation, under one condition, the claimed rang “collapses ... into an unusable void.” Opp. at 1-2; EX1039, ¶79. These allegations are wrong.

89. As background information, the ’711 patent describes placing different processing schemes into different groups. In particular:

- Group (i) processing schemes are “[h]igh cost (low refresh rate), high robustness algorithm[s]”;
- Group (ii) processing schemes are “[m]edium cost, medium robustness algorithms,” i.e., medium refresh rate algorithms; and
- Group (iii) processing schemes are “[l]ow cost (high refresh rate), low robustness algorithms.”

EX2019, ¶145; *see id.* ¶¶150-152 (listing different neural networks belonging to different groups).

90. The specification discloses:

[A] set of processing schemes (e.g., algorithms) are examined from group to group, and the *process cycles through the processing schemes* to produce a detection of an object. According to the example implementations, if the *confidence number is higher than an activation number*, an action is performed to stop or change a running group (i) and select or switch to group (ii). For example, a processing scheme from the group of (i) High cost (low refresh rate), high robustness algorithm can select a cheaper processing scheme from the group of (ii) Medium cost, medium robustness algorithms. Medium cost algorithms can be run from group (ii) in higher refresh rate, while monitoring the confidence level of the result.”

EX2019, ¶146. Thus, if the “confidence number,” i.e., the confidence level of the processing scheme that was just applied, is “higher than an activation number,” i.e., a high threshold, “a cheaper processing scheme,” e.g., a neural network having medium / low robustness but also medium / high refresh rate, is selected.

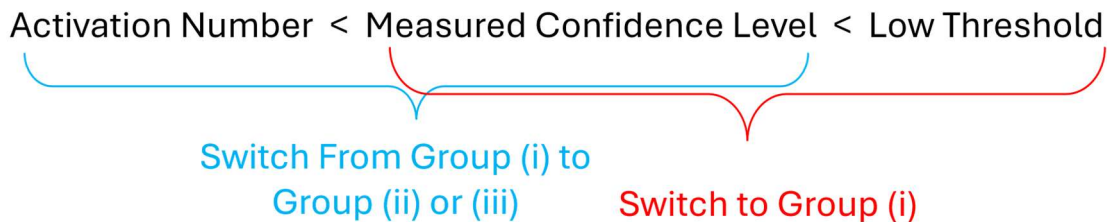
91. The specification also states:

If the confidence is *below low threshold*, falls outside of a *range*, ... group (i) processing is performed. If the *confidence is determined to be high*, then group (iii) algorithms can be applied to optimize resources.

EX2019, ¶147. Thus, if the confidence level falls below a “low threshold,” subsequent processing is performed using a Group (i), i.e., high robustness but low refresh rate processing scheme. If the confidence level is high, e.g., above the

“activation number,” a Group (iii) algorithm may be applied. It is described below that the “low threshold” is no greater than the “activation number.” As such, under both situations described above, the confidence level “falls outside of a range” defined by the “low threshold” and the “activation number.”

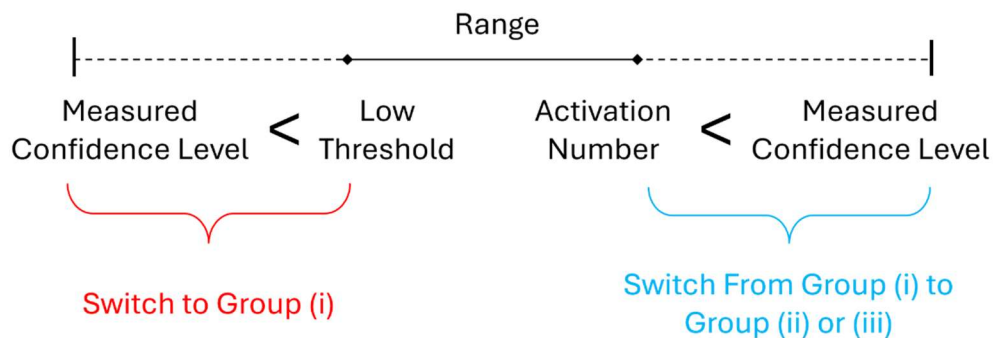
92. Specifically, to the extent the “low threshold” and the “activation number” are different, they form a range bounded by the low threshold, i.e., a *first threshold*, and the activation number, i.e., a high or *second threshold*. The above-discussed scheme for selecting a processing scheme from different group make it clear that the “activation number” must not be smaller than the “low threshold.” This is because commonsense dictates that such a condition would result in an inoperable, absurd system.



Demonstrative 3

93. In particular, as shown in the Demonstrative 3 above, both conditions that are described above would be true at the same time, causing the system to both switch away from Group (i) and switch to Group (i) at the same time. Thus, the disclosure of the '711 patent describes a scheme of thresholds where the “low threshold,” i.e., a *first threshold*, is less than the “activation number,” i.e., a *second*

threshold, as shown in Demonstrative 4 below. These thresholds define a range recited in SC 37. A new neural network selection is made if the confidence level “falls outside of” that range. *See* EX2019 ¶146. Should the value of the measured confidence level fall *within* the range, i.e., the value is greater than the “low threshold” but less than the “activation number,” no switching occurs, i.e., the neural network that was used to process the immediately preceding frame is continued to be used to process the current frame. The “otherwise” portion in each of claim elements [37.e3], [37.e4] indicates this behavior.

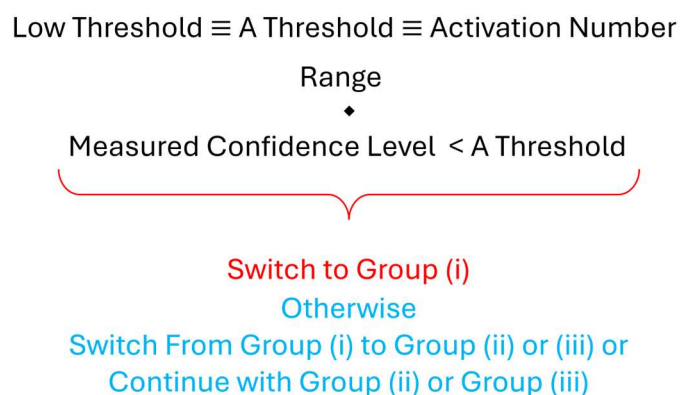


Demonstrative 4

94. The '711 patent also discloses an active process switching scheme that uses only one threshold. EX2019, ¶112-113 (“[i]n response to the performance score of the active processing scheme failing to satisfy *a threshold*, the method selects another processing scheme to process the video data stream” where “*the threshold* can be based on available computing resources associated with the mobile capture device”).

95. In this case, the low threshold and the activation number are replaced by or are both equal to a single threshold, and the range collapses essentially to a single point, but not “into an unusable void,” as Dr. Singhose erroneously alleges. I am informed by Counsel and understand that if one claim limitation causes another to reduce to a single point, that does not cause the claim to be indefinite. *Maxell, Ltd. v. Ampere Technology Ltd.*, 94 F.4th 1369, 1372-73 (claim elements were not contradictory, and hence, claim was not indefinite, even though one claim limitation reduced another limitation to a single point).

96. When the low and high thresholds are the same, it follows logically and incontrovertibly that if the measured confidence level is below the single threshold, Group (i) processing scheme is selected. Otherwise, if the last processing scheme were from Group (i), a “cheaper” processing scheme from Group (ii) or (iii) is selected, and if the last processing scheme were already from those groups, processing continues with the selected processing scheme. This is shown in Demonstrative 5 below.



Demonstrative 5

97. Accordingly, the specification fully supports “*a first threshold of a range defined by the first threshold and a second threshold greater than or equal to the first threshold*” as claim element [37.e3] recites.

3. Specification Discloses Applying Second Neural Network When Confidence Is Higher than Second Threshold

98. Claim element [37.e4] recites “applying the second neural network to the current image when the confidence level is above the second threshold,” where, per claim element [37.e3], “a range [is] defined by [a] first threshold and [the] second threshold greater than or equal to the first threshold.” Petitioners allege that “confidence level, and using it in this way, is not supported as in [37.e4].” Opp. at 2; EX1039, ¶81. This allegation and additional related allegations that Dr. Singhose makes are false.

99. I have explained above that the specification fully discloses the use of: (i) two thresholds, a “low” first threshold and an “activation number,” i.e., a higher second threshold, defining a range; and (ii) a single threshold, where the first and second thresholds are the same as the single threshold, and the range becomes a single point, i.e., the single threshold. *See supra* § VII.A.2.

100. The specification states that “if the confidence number is *higher than an activation number*,” i.e., the second threshold, an action is performed to stop or change a running group (i) and select or switch to group (ii).” EX2019, ¶¶146, 147

(“[i]f the confidence is determined to be high, then group (iii) algorithms can be applied to optimize resources). As described in the RMTA, the first neural network is selected from Group (i) and the second neural network is selected from Groups (ii) or (iii). *See* RMTA at 22. Thus, the specification discloses “applying the second neural network to the current image when the confidence level is above the second threshold,”

B. Specification Fully Supports SCs 21 and 24

1. Written Description Supports Identifying a Model Type Based on Target Classification

101. SC 24 recites “identifying a model type of the target based on the classification of the detected target.” Petitioners state that “the specification describes classifying by type or model, [but] not identifying type based on [a] classification,” and argue that as such, SC 24 lacks written description support. *Opp.* at 2; EX1039, ¶83. To the extent this argument requires SC24 to perform a target classification first and then determine the model type using the determined classification, such an interpretation is erroneous.

102. Claim element [21.b] recites “identify a classification of the detected target,” and SC 24 recites “identifying a model type of the target based on the classification.” But these are not necessarily two separate steps in a series that must be performed sequentially. They are two different kinds of classifications, the one

per SC 24 being narrower than the one per claim element [21.b]. In particular, the specification states:

The one or more machine learning models 205 may be trained using a set of training data. The set of training data includes a set of positive examples and a set of negative examples. For example, the set of positive examples may include a plurality of feature vectors that indicate the detected object is a UAV. The set of negative examples may include a plurality of feature vectors that indicate the detected object is not a UAV. For example, the set of negative examples may include feature vectors associated with a balloon, bird, plane, helicopter, etc.

EX2019 ¶50; EX1001, 7:32-41 (a first kind of classification).

103. The specification also states: “[t]he output of the image analysis (e.g., *output from applying the machine learning model*) may include ... a classification of the detected target (e.g., *type, model, class, manufacturer, size, etc. of the detected target*).” EX2019 ¶94; EX1001, 17:25-38 (a second kind of classification).

104. Thus, the first kind of classification may only classify a detected object as a UAV or not a UAV, and the second kind of classification may classify the detected object according to its model type, i.e., both model and type. The model type per SC 24 is determined using the second kind of classification, as shown in the RMTA. *See* RMTA at 9 (quoting EX2019 ¶94). In my view, the

Board has performed substantially the same analysis and has reached the same conclusion. *See* Paper 30 (PG) at 6-7. As such, the specification supports SC 24.

2. Written Description Teaches Two Speed Adjustment Factors for SCs 21 and 25, Respectively

a Overview of Speed Adjustment Factors

105. The specification discloses several speed adjustment factors that are claimed in various SCs. I provide an overview of these factors to aid the discussion below. According to the specification, “a vector identifying the relative location of the target is determined using at least the determined relative direction of the target.” EX2019, ¶97. “In some embodiments, the vector is defined by three-dimensional coordinate-based values” that represent the target’s relative location. *Id.* According to the specification, the vector may also be defined by “a *factor* based on an associated distance or speed” where a determined distance to the target “is mapped to a *desired speed* of travel to reach the target.” *Id.* The vector may also include an “acceleration factor.” *Id.* ¶99. Thus, a “factor” is a vector element that represents a value or a *desired value*. that may be used in controlling the operation of a UAV A “speed adjustment factor” is a vector element value used to control, i.e., adjust speed.

106. During flight, the value of an attribute of the UAV, e.g., speed or acceleration, that was desired at one time may be changed or adjusted at another time. This entails adjusting the factor corresponding to the attribute to be changed.

See id. ¶99 (“[m]odifying the speed and/or acceleration associated with the vector includes *adjusting a factor used to control the speed* and/or acceleration of the interceptor aerial vehicle”). Upon such an adjustment of the factor, the factor, in turn, represents an adjustment to be made in controlling the corresponding UAV attribute (e.g., speed or acceleration). Thus, a factor that can represent adjusted attribute values is an adjustment factor.

107. A factor can be adjusted in two ways: by scaling, i.e., by multiplying the current value of the factor by a scalar value; or directly, e.g., by computing a new value based on some other values. For the speed attribute, the speed may be changed in at least two different ways: (i) based on a magnitude of deviation of a direction of the three-dimensional relative location with respect to a current flight direction of the reference aerial vehicle, *see Id.* ¶99; and/or (ii) based on the determined distance to the target. Both types of changes may be desired at the same time, and such, two instances of the speed adjustment factor may be used, each representing a corresponding adjustment. *See Id.* ¶97.

108. To make the first change, “the speed and/or acceleration factor is multiplied by one or more scalar values that change in magnitude based on a directional deviation of the vector from a current direction of movement.” *Id.* ¶99. Thus, the value represented by a speed factor, i.e., a speed adjustment factor because it represents a speed adjustment, is adjusted by multiplying the initial

value by a scalar that changes based on the direction deviation. To make the second change, “the vector is at least in part defined,” i.e., the speed adjustment factor vector element in the vector is defined “using a distance and/or speed value,” where “the distance value determined [based on image analysis] is *mapped to a desired speed* of travel to reach the target (e.g., greater speed desired for further distance and slower speed desired for shorter distance to allow more time for direction correction).” *Id.* ¶97.

109. Finally, the specification states:

a scaled three-dimensional vector based on the relative location of the target and determined based on the detected image of the target is used to *adjust a flight path* of the interceptor aerial vehicle in the direction and intensity of the three-dimensional vector. For example, the three-dimensional *vector specifies* directional and *speed/acceleration changes* to be performed by the interceptor aerial vehicle to direct the flight path of the interceptor aerial vehicle

Id. ¶91; *see id.* ¶101 (“the scaled [i.e., adjusted] vector and associated directional values and distance, *speed*, and/or acceleration values are provided for use in navigating the interceptor aerial vehicle towards the target aerial vehicle”) *Id.*

¶101. As the desired speed changes from time to time during flight, the speed adjustment factor can be modified and/or redefined at different times such that the first speed adjustment factor is provided at one time and the second speed adjustment factor is provided at another time.

b First and Second Speed Adjustment Factors of SCs 21 and 25

110. In connection with SC 25, Petitioners argue that the specification does not support determining first and second speed adjustment factors that are *both* based on the determined distance to the target. Opp. at 2; EX1039, ¶85. This argument is erroneous; it ignores a related limitation that claim element [21.d] recites.

111. In particular, claim element [21.d] recites “determining a *first* speed adjustment factor *based on a magnitude of deviation*” and SC 25, which depends from SC 21, recites “determining a *second* speed adjustment factor *based on the determined distance to the target.*” Thus, SC 25 does require determining two different speed adjustment factors, but they *both are not* based on the determined distance to the target. Specifically, SC 25 does not require a first speed adjustment factor that is based on a determined distance to the target and a second speed adjustment factor that is also based on a determined distance to the target. Rather, per SC25, which depends from SC 21, the first factor is based on a magnitude of deviation and the second factor, is based on the determined distance to the target.

112. Dr. Singhose also alleges that the cited “disclosures do not even contend that there is a second speed adjustment factor in addition to the first speed adjustment factor.” EX1039, ¶85. This allegation is false. If it is based on an interpretation that the first and second speed adjustment factor are *both* based on a

distance, that interpretation is incorrect as described above. SC 25 does not recite and does not require two speed adjustment factor that are *both* based on a distance.

113. I have explained above that the specification discloses a speed adjustment factor, the value of which may be adjusted by a scalar, where the magnitude of the scalar changes based on a direction deviation. *See supra* §VII.B.2.a; EX2019, ¶99; *see* RMTA at 6-8. Thus, the specification describes the “first speed adjustment factor” that claim element [21.d] recites. I also explained that the speed adjustment factor in the vector is defined “using a distance and/or speed value,” where “the distance value determined [based on image analysis] is mapped to a desired speed of travel to reach the target.” *Id.* ¶97; RMTA at 9-10; *see also supra* §VII.B.2.a (discussing two instances of the speed adjustment factor). Thus, the specification also discloses the “second speed adjustment factor” that SC 25 recites. The specification discussed herein is cited in the RMTA, and as such, those cited portions of the specification teach the speed adjustment factors that SCs 21 and 25 recite.

VIII. PSCs 21-37 ARE PATENTABLE

A. SCs 21-37 Are Enabled by the Specification

1. Claimed Speed Adjustment Factors Are Enabled

114. Petitioners state that SCs 21, 36, and 37 recite determining a “speed adjustment factor,”⁵ and argue that SCs 21, 36, and 37 are not enabled because a POSITA allegedly “cannot practice the SCs without undue experimentation.” Opp. at 2-5; EX1039, ¶¶86-89; *see id.* ¶¶90-101. This and other arguments Petitioners present in connection with this claim limitation are incorrect. Before I explain why, I note that Dr. Singhose’s related arguments are internally inconsistent and he and Petitioners allege facts that are clearly untrue.

115. For instance, citing par. 99 of the ’452 App. Dr. Singhose states that specification describing a speed adjustment factor describes “go slower when turning hard, go faster when flying straight,” but “says nothing about how to determine a speed adjustment factor.” EX1039, ¶90. Petitioners allege that “[no] formula” is provided to determine a speed adjustment factor, Opp. at 3, and later, referring to Equations 7 and 8 of the specification, argue that actual values for the constants for a “formula” and not assigned actual numbers. Opp. at 3-4; EX1039, ¶92.

⁵ SC 21 recites “a *first* speed adjustment factor.”

116. The specification fully discloses the speed adjustment factors recited in SCs 21, 36, and 37 (and also in SCs 25 and 31). *See supra* § VII.B.2.a; EX2019, ¶¶98-99 (discussing “smoothing and limiting the rate of change associated with the vector” providing the speed adjustment factors). Thus, the meaning of a “speed adjustment factor” is clear from the specification, and a POSITA would have understood that a “speed adjustment factor” is a parameter or a value at a particular instance of time that is used to control, e.g., increase or decrease, and thus adjust, speed.

117. To the extent a speed adjustment factor simply adjusts the speed such that the RAV goes slower while turning hard and faster while flying straight, a speed adjustment factor can be any factor such as 0.8 that would decrease the speed to 80% of the current speed or 1.25, that would increase the speed by 25%. A POSITA would understand to choose a factor such that the minimum and maximum speed limits for the UAV are not violated. I am informed by Counsel and understand that further disclosure is “preferably omit[ted].” *McRO, Inc. v. Bandai Namco Games America Inc.*, 959 F.3d 1091, 1102 (Fed. Cir. 2020) (“a ‘patent need not teach, and preferably omits, what is well known in the art’”) (quoting *Spectra-Physics, Inc. v. Coherent, Inc.*, 827 F.2d 1524, 1534 (Fed. Cir. 1987).)

118. Petitioners noted that the speed adjustment factor limitation was introduced during prosecution, may therefore be a novel aspect of the claimed invention. Opp. at 2. Accordingly, the specification does describe various specific relationships that would inform a POSITA how a speed adjustment factor can be selected and used. The specification also provides an equation, which Petitioners discuss, while also falsely alleging that the specification provided no formula. On the one hand, according to Dr. Singhose, the speed adjustment factor simply informs “go slower when turning hard, go faster when flying straight,” but on the other hand, Petitioners’ allege that undue experimentation is necessary because specific curves and the constants of an equation are not revealed. Notwithstanding these inconsistencies and factual error, I performed *In re Wands* analysis, as discussed below.

119. Starting with **Factors 2 and 3**: “the amount of direction or guidance presented” and “the presence ... of working examples,” is adequate. For the speed adjustment factor, a POSITA would have readily known that if the target is farther away, the speed should be increased. Nevertheless, the specification describes this. EX2019, ¶97 (“greater speed desired for further distance and slower speed desired for shorter distance to allow more time for direction correction”); EX1001, 18:54-59. The specification also states that “when a large directional change is required ... a slower speed is desirable to allow the directional change to take place at a

slower speed over a shorter distance as compared to a faster speed that is desirable to reach the target quicker when no or minor directional change is required.”

EX2019 ¶¶99; EX1001, 19:8-14.

120. The specification also states that “a vector identifying the relative location of the target” may be “defined using a distance and/or speed value.” EX2019, ¶¶97; EX1001, 18:44-59. “Modifying the speed ... includes adjusting a factor used to control the speed” and the factor may be “multiplied by one or more scalar values.” EX2019, ¶¶99; EX1001, 19:4-20. Thereafter, the specification discloses the relationship between the scalar that modifies the speed adjustment factor and “a directional deviation” as a curve, e.g., a “bell curve” or a linear curve. EX2019, ¶¶99; EX1001, 19:20-29. The specification also provides Equations 7, 8 for determining “speedfactor[s],” i.e., the speed adjustments. EX2019, FIG. 10.

121. Petitioners argue that the disclosure of the “scalar” is unclear because the specification only describes a curve shape and “not how to actually determine the scalar.” Opp. at 3; EX1039, ¶¶91. Petitioners further argue that the constants in Equations 7 and 8 are not assigned “actual numbers.” Opp. at 4; EX1039, ¶¶91. In my opinion, providing one or some specific curves or specific values of the equation parameters is not particularly helpful.

122. This is because, RAVs in many different shapes and sizes are available, and their intended usage, even when operating as an interceptor, varies

significantly. For example, the operation of an interceptor is designated to protect the airspace over a sports stadium is different from that designated to protect the airspace of an airport. On the basis of a selected curve shape, selecting a particular curve, or choosing the specific constants of the equation used to compute the speed adjustment factor is an optimization problem that would vary substantially from RAV to RAV and from one usage scenario to another. As such, any specific curves and/or equation constant values provided in the specification would generally not be applicable or even meaningful to RAVs at large.

123. On the other hand, such curve and equation constants selection can be readily accomplished when the specifics about the RAV (e.g., its size, weight, shape, maximum speed, maximum altitude, etc.) and specifics about the environment in which the RAV would operate, e.g., the type of GPU, the nature of the terrain (e.g., mountainous, wooded, urban, etc.) are known and taken into account. The use of this information in choosing specific curves / constant values would be highly beneficial, and may even be invaluable. This would involve only a reasonable amount of experimentation. Also, regarding Equations 7, 8, a POSITA would have readily recognized those as corresponding to a type of sigmoid functions, the use of which for smoothing variables, e.g., those representing changes (change in speed in this context) is well known. Coincidentally, sigmoid functions are used extensively in the design of artificial neural networks. As such, a

POSITA would have recognized the function of the equation constants, and how to select them via a reasonable amount of experimentation.

124. I am informed by Counsel and understand that the experimentation noted above may be needed is consistent with the enablement requirement. *PPG Industries, Inc. v. Guardian Industries Corp.*, 75 F. 3d 1558, 1564 (Fed. Cir. 1996) (the “fact that some experimentation is necessary does not preclude enablement” as long as “the amount of experimentation” is not “unduly extensive”). Moreover, the determination of a specific curve and equation parameters is a production-level detail. I am informed by Counsel and understand that disclosure of such details is not required for enablement. *Koito Mfg. Co., Ltd. v. Turn-Key-Tech, LLC*, 381 F.3d 1142, 1155 (Fed. Cir. 2004) (without evidence to the contrary, “such production details [are not] necessary to practice the claimed invention without undue experimentation”). Thus, in my opinion the specification provides substantial guidance and/or examples.

125. **Factor 1:** “quantity of experimentation necessary,” is not high, as discussed above. Moreover, while the inventions are directed to on-board a UAV application of ML models and neural networks for efficient image analysis, the selection of speed adjustment factors is related to the conventional mechanics of UAV flight. Therefore, the experimentation necessary to determine these

adjustment factors was routine. As such, a high quantity of experimentation may be tolerated. *In re Wands*, 858 F.2d at 737.

126. **Factor 4:** “the nature of the invention,” is complex. The innovative aspects include: (i) an NN that can both provide a bounding box for a target in an image and identity a type of the target, using which a physical size of the target can be obtained, which allows for more accurate determination of the three-dimensional target location (per SC 21); (ii) the use of two NNs in parallel, where a high-robustness NN performs target detection, which can be accurate, and a low-robustness NN performs incremental prediction of target location, which can be faster (per SC 36); and (iii) the use of one of two NNs, where a low-robustness, high refresh rate NN is used when the confidence level of target detection is high, and a high-robustness, low refresh rate NN is used when the confidence level is low. But as described above, the selection of the speed adjustment factors is related to conventional mechanics of UAV flight.

127. **Factors 5 and 7:** “the state of the prior art” and “the predictability or unpredictability of the art,” also suggest that undue experimentation would not be needed. For example, as the Petition states, Pet. at 42-42, 62-64, Beard, Pierce, and Sahasrabudhe teach speed adjustment factors. EX1009, 3:25-30 (“*adjust the speed or position of UAV 100 and maintain the initial orientation (e.g., keep a skier centrally framed at a fixed distance as s/he proceeds through a downhill run)*);

EX1025, ¶17 (the “necessary correction,” i.e., a speed adjustment factor, to “move the follower aircraft” “within an acceptable range” from a lead aircraft “is determined and transformed into body-axis velocities”).

128. Per **Factor 6**: “the relative skill of those in the art,” is high, which suggests only a reasonable amount of experimentation would be needed. **Factor 8**: “the breadth of the claims,” is rather narrow, because per SC 21, an ML model that can provide: (i) a bounding box surrounding the detected features of a detected target, and (ii) classifying the type of the target is needed. The scope of SCs 36 and 37 is also rather narrow, because each claim requires two neural networks, having different levels of robustness and cost / refresh rate, and selective execution of those neural networks.

129. In sum, in my opinion, *In re Wands* analysis shows that undue experimentation is not needed to determine and use the claimed “speed adjustment factor,” and as such, these limitations are enabled.

2. Claim Limitations Robustness, Compute Cost, and Refresh Rate Are Enabled

130. SCs 36 and 37 recite claim terms first and second robustness, first and second compute cost, and first and second refresh rates. Petitioners argue that claim elements reciting these limitations are not enabled. Opp. at 5-8; EX1039, ¶104 (arguing that a “POSITA looking for guidance on how to determine any of these values would find nothing actionable in the specification”); *see* Opp. at 6;

EX1039, ¶106 (arguing that the specification does not identify the unit of compute cost); Opp at 7; EX1039, ¶ (arguing that a POSITA is not informed “about how to determine, quantify, or assign a refresh rate to any particular processing scheme”).

131. Each of these arguments fails because before the priority date of the ’711 patent the attributes robustness, compute cost, and refresh rate, of a neural network were well-known. Moreover, the values of these attributes for different types of NNs were published, and were available to a POSITA. Furthermore, SCs 36 and 37 do not recite any particular target values for any of these attributes. Rather, these claims only require one NN to have a higher or lower attribute value than that of another NN. To facilitate such choice, the specification explicitly identifies the NNs that have relatively higher / lower attribute values. As such, even if the actual attribute values of those NNs are not known, the relative ordering provided in the specification clearly informs a POSITA which NN(s) can be used as the claimed first neural network and which NN(s) can be used as the claimed second neural network.

132. The term robustness in the context of a neural network (and a processing scheme, in general) has the same meaning as in other contexts – susceptibility to adverse conditions, e.g., noise, in the case of image analysis. A POSITA would have known that robustness can be determined in terms of failure rate or frequency. The literature discussing neural networks used the term

“robustness” in this way. *See e.g.* EX2047 at 7 (p. 2016); EX2048 at 9 (p. 1937); EX2042, Abstract.

133. Compute cost is simply the cost of computations. In general, this cost can be expressed using many parameters, individually, or in combination. For example, compute cost can simply be the total number of computations, or the number of floating-point computations. It can also be the size of hardware and/or software used for the computation, power consumed by the computation, and time taken by the computation. The '711 patent associates cost with robustness and refresh rate with clarity and consistency. *See* EX2019 ¶¶145, 146, 150, EX1001, 27:53-64, 28:5-8, 28:40-29:3. More specifically, the '711 patent discloses that a high cost is associated with high robustness, and low refresh rate, while a low cost corresponds to low robustness, and high refresh rate. *See id.* Dr. Singhose agrees. EX2033 at 37:6-16.

134. A POSITA would have known that refresh rate means the rate at which new data can be analyzed, e.g., the rate at which new images or frames of a video can be analyzed. The literature discussing neural-networks for image analysis used this term in this way. *See e.g.* EX2049 at 5; EX2050 at 2 (p. 12), Figure 2, at 8 (p. 18). A POSITA would have also known that a low refresh rate means the processor requires more compute time for processing each image / frame (a piece of data, in general). Thus, the cost of the processing in terms of

compute time is high. This is exactly the relationship between cost and refresh rate the '711 patent describes. *See* EX2019 ¶¶145, 146, 150, EX1001, 27:53-64, 28:5-8, 28:40-29:3.

135. Thus, a POSITA would have clearly understood, in fact known, the meanings of each of the terms discussed above because they were described in the specification in the same way as in published art related to neural networks and image processing. *In re Wands*, 858 F.2d 737. Therefore, without more a POSITA would have also known how to determine these quantities to the extent such a determination is necessary. *McRO*, 959 F.3d at 1102 (“a ‘patent need not teach, and *preferably omits*, what is well known in the art”). The '711 patent nevertheless provides further guidance, as I discuss further, analyzing the eight *In re Wands* factors.

136. Starting with **Factors 2 and 3**: “the amount of direction or guidance presented” and “the presence ... of working examples,” is high because the '711 patent provides specific, representative neural networks in connection with high / low robustness, high / low cost, and high / low refresh rate. *See* EX2019 ¶¶150-152; EX1001, 28:40-29:3. For instance, the '711 patent identifies “tiny” YOLO as a high robustness neural network and SqueezeNet super fast rejection CNN as an example of a low robustness neural network. The word “tiny” is not a generic descriptor; it is a specific type of YOLO. *See* EX2045 at 1-2 (pp. 1935-36)

(distinguishing tiny YOLO from YOLO); EX2051 at 1-2 (same); EX2052 at 2 (same). Therefore, a POSITA would have understood that any neural network that has a higher cost, e.g., in terms of computation time, than tiny YOLO would not be a low cost or robustness or high refresh rate neural network.

137. Additionally, a POSITA would have known that in general, the larger the size of the image to be processed, the greater the computation time / cost of a neural network. The '711 patent discloses specific image sizes, EX2019 ¶¶150, 151; EX1001, 28:40-53, although neural networks can be used to process images having substantially smaller or larger sizes than those disclosed in the '711 patent. *See e.g.* EX2053 at 7 (p. 162). Thus, for all of the neural network attributes that Petitioners identified, the specification provides substantial guidance and/or examples.

138. **Factor 1:** “quantity of experimentation necessary,” is not high because the '711 patent provides specific examples of neural networks and image sizes that may be used as high and low robustness neural networks. Moreover, while SCs 36 and 37 are directed to *scheduling* high and low cost, low and high refresh rate neural networks, they do not require development of new neural networks. Rather, NNs for particular operations (object detection or incremental prediction per SC 36) or based on the confidence level (per SC 37) are chosen from neural networks that are known. Researchers and practitioners in the field routinely

reported attributes such as robustness, computation cost (time), and refresh rate of neural networks. *See* EX2047 at 7 (p. 2016); EX2048 at 9 (p. 1937); EX2042, Abstract; EX2049 at 5; EX2050 at 2 (p. 12), Figure 2, at 8 (p. 18). As such, the experimentation necessary to determine these attributes of a neural network was routine. This would allow a high quantity of experimentation. *In re Wands*, 858 F.2d at 737.

139. **Factor 4:** “the nature of the invention,” is complex as discussed above. But, here again, the disclosed specific examples of neural networks and explicit guidance on their relative robustness, cost, and refresh rates can effectively help navigate the complexities of making and using the claimed systems without undue experimentation. **Factors 5 and 7:** “the state of the prior art” and “the predictability or unpredictability of the art,” also suggest that undue experimentation would not be needed. This is because several different types of neural networks and their characteristics in terms of robustness, execution speed, accuracy were known and published. *See e.g.* EX1036 at 48 (p. 779); EX2054 at 7-8 (p. 2350-51); EX2047 at 7 (p. 2016).

140. Per **Factor 6:** “the relative skill of those in the art,” is high, which suggests only a reasonable amount of experimentation. **Factor 8:** “the breadth of the claims,” is rather narrow. Each of SCs 36 and 37 requires two neural networks,

having different levels of robustness and cost / refresh rate, and conditional or selective execution of those neural networks.

3. Claim Limitations First Threshold, Second Threshold, and Range Are Enabled

141. Petitioners argue that the specification does not disclose “a first threshold and second threshold, defining a range, set equal.” Opp. at 8-9; EX1039, ¶119. This is not true. I have explained above that the specification fully discloses the use of: (i) two thresholds, a lower threshold and a higher threshold, defining a range; and (ii) a single threshold, where the lower and upper thresholds are the same as the single threshold, and the range becomes a single point, i.e., the single threshold. *See supra* § VII.A.2.

142. Thus, contrary to Petitioners’ arguments, the specification clearly describes the claimed range and thresholds, including the condition where the second threshold is equal to the first threshold. An application of the *In re Wands* factors shows that this disclosure is enabling.

143. Starting with **Factors 2 and 3**: “the amount of direction or guidance presented” and “the presence ... of working examples,” is adequate. As a starting point, the above-described process selection scheme is similar to systems in use every day. For example, modern home thermostats have two threshold settings – one to determine when to turn on heating and the other to determine when to turn

on cooling.⁶ Climate control systems in cars often feature setting only a single target temperature. In light of this common knowledge, the guidance from the specification is substantial, as discussed above. *See supra* ¶¶141-142.

144. Moreover, as in the case of the actual values of the constants of the equations provided for determining the speed adjustment factors, the actual threshold values are production-level detail. This is because, to determine the thresholds effectively, examining the confidence levels reported by the NNs while operating in the intended environment is instrumental, if not essential. Values obtained in one environment are generally not directly useful in another environment. Once the confidence levels are observed, however, from a reasonable amount of experimentation, the thresholds can be readily selected based on a target frame processing rate, which would limit the time spent in processing by the high cost / low speed neural network.

145. **Factor 1:** “quantity of experimentation necessary,” is minimal. Based on the confidence levels that neural networks report, any suitable threshold values (where range is defined by those thresholds) or a single threshold can be selected to obtain the benefit of the invention of SC 37. The selection of the best thresholds

⁶ I note that a thermostat is not identical to the process switching system. For example, the threshold to turn on cooling is strictly greater than the threshold to turn on heating, to avoid thrashing.

or single threshold would require some experimentation, but such an optimization is not claimed, and is a production-level detail that need not be disclosed in the specification. *Koito*, 381 F.3d at 1155. **Factor 4:** “the nature of the invention,” is complex. But, due to the limited number of choices to be made, and as also discussed for Factor 1, only a reasonable amount of experimentation may be needed. **Factors 5 and 7:** “the state of the prior art” and “the predictability or unpredictability of the art,” do not have much of bearing on the amount of experimentation needed. While switching between different types of neural networks, as described and claimed in the ’711 patent is novel, the threshold selection is mainly an optimization problem. The switching between two neural networks that would occur even when sub-optimal thresholds are used would result in performance improvement as the ’711 patent describes.

146. Per **Factor 6:** “the relative skill of those in the art,” is high, which suggests only a reasonable amount of experimentation. **Factor 8:** “the breadth of the claims,” is rather narrow. SCs 37 requires two neural networks, having different levels of robustness and cost / refresh rate, and conditional or selective execution of those neural networks.

147. In summary, for the speed adjustment factor, the robustness, compute cost, and refresh rate attributes, and the switching thresholds, which define the range, all of the *In re Wands* factors demonstrate that in light of the specification,

no undue experimentation would have been necessary to make and use the invention per SCs 21, 36, and 37. As such, in my opinion, SCs 21-37 are enabled. Moreover, the provided guidance and examples would have informed a POSITA what the inventor had invented and that the inventor was in possession of the claimed invention.

B. SCs 21-37 Are Not Indefinite

1. Terms Common to SCs 21, 36, and 37 Are Not Indefinite

148. In arguing indefiniteness, Petitioners state that various claim elements recite the following terms: “a speed adjustment factor,” “first robustness,” “second robustness,” “first compute cost,” “second compute cost,” “first refresh rate,” and “second refresh rate.” They allege that “it is unclear ... what the terms mean” and that a “POSITA would not have known their boundaries.” Opp. at 9; EX1039, ¶¶125. These allegations are false.

149. The meanings of the speed adjustment factors recited in SCs 21, 36, and 37 (and in SCs 25 and 31) would have been clear to a POSITA, who would have understood that a “speed adjustment factor” is a vector element value used to control, i.e., adjust speed. *See supra* § VII.B.2. The meanings of the terms robustness, compute cost, and refresh rate would have also been clear because, as I discussed above, these terms were well understood, the specification uses them in

the same way in which the literature used them, and provided guidance and examples explaining use and meaning of these terms. *See supra* § VIII.A.2.

150. Referring to the comparative relationship between the robustness, compute cost, and refresh rate terms, Petitioners argue that “[w]ithout knowing their boundaries/how they are determined or measured, there is no way of comparing them to determine such a relationship.” Opp. at 9; EX1039, ¶127. This argument fails for several reasons.

151. The meanings of the related limitations recited in SC 36 and 37 are clear. “[N]o way of comparing” the attributes recited in these claims is not an indefiniteness but a written description support and/or enablement issue. I am informed by Counsel and understand that “if the ‘enabling’ disclosure of a specification is not commensurate in scope with the subject matter encompassed by a claim, that fact does not render the claim imprecise or indefinite or otherwise not in compliance with [§ 112(b)]; rather, the claim is based on an insufficient disclosure.” *In re Hyatt*, 708 F.2d 712, 714-15 (Fed. Cir. 1983); *Process Control Corp. v. HydReclaim Corp.*, 190 F. 3d 1350, 1358, n.2 (Fed. Cir. 1999) (“definiteness and enablement are analytically distinct requirements”). I have already discussed these claim terms and have explained how they can be determined, and that they are enabled. *See supra* § VIII.A.2; EX2047 at 7; EX2048 at 9; EX2042, Abstract; EX2049 at 5; EX2050 at 2, Figure 2, at 8.

152. Regardless, the specification states:

Costs on all data inputs for the processing schemes can be pre-determined, estimated, or calculated. For example, a type of algorithm that operates on visual “features” can be dependent on the number of features in the video frame, and its cost [may be] estimated based on the number of features. As the number of features increases the cost can increase dramatically.

EX2019, ¶142; EX1001, 27:36-42. The ’711 patent also describes that a high cost processing scheme has a low refresh rate and that a low cost processing scheme has a high refresh rate. EX2019 ¶¶145-150; EX1001, 27:53-28:47. Dr. Singhose agrees. EX2033 at 37:6-16. Thus, obtaining the respective costs of two neural networks, e.g., according to the reported values as described above, and/or based on the average number of visual features to be processed, the relative ordering of those two neural networks on the basis of their robustness and/or refresh rates would also be readily discernable.

153. Additionally, the specification explicitly identifies neural networks that meet the comparative relationships. To this end, the specification categorizes the different neural networks into three groups: Group (i) processing schemes are “[h]igh cost (low refresh rate), high robustness algorithm[s]”; Group (ii) processing schemes are “[m]edium cost, medium robustness algorithms,” i.e., medium refresh rate algorithms; and Group (iii) processing schemes are “[l]ow cost (high refresh

rate), low robustness algorithms.” EX2019, ¶145. Then, for each group, the specification identifies at least one neural network belonging to that group. *See id.* ¶¶150-153. Thus, even without knowing the actual boundaries, i.e., the robustness, compute cost, and/or refresh rate values, a POSITA would be informed on how two neural networks are related with respect to an attribute of interest, for at least some neural networks.

154. Accordingly, the above-identified terms that are common to SCs 21, 36, and 37 do not render the claim elements reciting these terms and dependent claims of SC 21 unclear or indefinite.

2. Limitations Recited in SC 37 Are Not Indefinite

155. In arguing that SC 37 is indefinite, Petitioners argue that “if no network has yet been applied to the current image, the system cannot have a confidence level for a detection of the target in the current image, rendering the claim indefinite.” Opp. at 9-10; EX1039, ¶129-33. As I explained above, this argument is based on a flawed interpretation of claim element [37.e1].

156. The “confidence level *for* a detection ... in the current image” is not a confidence level *based on* or *derived from* a detection in the current image. Rather, the confidence level obtained from the detection of a target in the immediately preceding image informs the confidence in how good a detection in the current image would be (i.e., confidence *for* a detection) if the same processing scheme

that were used to process the immediately preceding image were also used to process the current image. *See supra* ¶¶66. As such SC 37 does not have a circular dependence – requiring the processing of the current image to determine which processing scheme is to be used to process the current image.

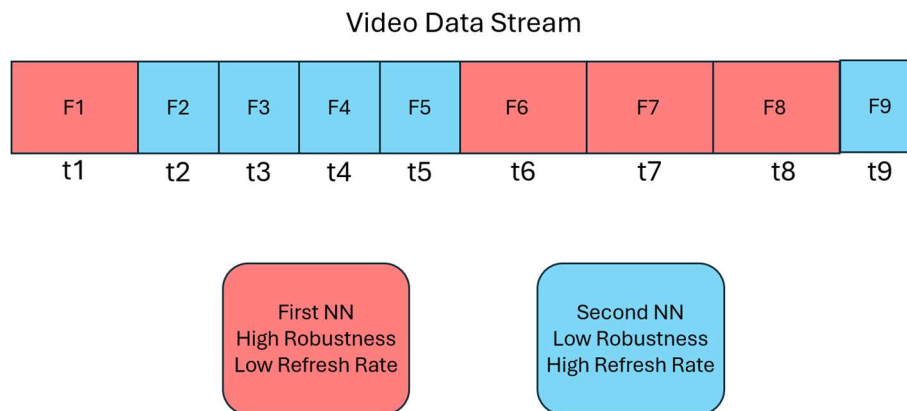
157. Petitioners further argue that the “specification never discusses ... an image-by-image processing technique.” Opp. at 10; EX1039, ¶134. This is incorrect, as explained above. *See supra* § VII.A.1.

158. Finally, Petitioners argue that “[a]s SC 37’s first and second NNs operate at different refresh rates, the immediately preceding image processed by the first NN may not be the same temporal frame as the immediately preceding image processed by the second NN.” Opp. at 10; EX1039, ¶134. This argument is flawed.

159. To the extent this argument supposes that two immediately preceding frames could exist simultaneous, SC 37 precludes that possibility. SC 37 recites that “detecting of the target in the current image further comprises making a selection, from between the first and the second neural networks, of *only one of* the first and the second neural networks.” It follows that each current image is the immediately preceding image of the image immediately following the current image. Thus, two neural networks are not applied to the same image. They are applied in a mutually exclusive manner. Therefore, for any current image, there is

only one immediately preceding image regardless of which neural network were used to process that immediately preceding image. The above assertion thus fails.

160. Petitioners argument fails without the supposition noted above as well. Naturally, a high refresh rate neural network generally processes frames / images at a faster rate than a low refresh rate neural network. But the varying rates do not affect the image-by-image processing described above, as seen in the Demonstrative below.



Demonstrative 6

161. In the Demonstrative above, different frames are processed by one of two neural networks, each having a different refresh rate. Accordingly, the respective processing times for different frames are different. It should be noted that the frames are all of the same size and are not smaller or larger. Demonstrative 6 depicts them as narrower and wider to indicate their processing times – not their sizes.

162. In Demonstrative 6, upon processing frame F2 at time t2 using Second NN, based on the confidence from processing F2, it is determined that Second NN will continued to be used to process frame F3 at time t3. Similar decisions are made for frames F3-F5. For instance, based on the confidence from processing F4 at time t4, it is determined that Second NN will continued to be used to process frame F5 at time t5. The confidence level obtained from processing F5 causes a different, slower neural network, i.e., First NN, to be selected for the processing of frame F6. As such, F6 is processed at time t6 at a slower rate.

163. The decision to use First NN for F6 does not depend however, on the faster processing rate of Second NN and the resulting faster processing of frames F2-F5. The confidence level obtained from processing F8 causes a different, faster neural network, i.e., Second NN, to be selected for the processing of frame F9. As such, F9 is processed at time t9 at a faster rate. But the decision to use Second NN for F9 does not depend on the slower processing rate of First NN and the resulting slower processing of frames F6-F8.

164. Thus, the different refresh rates of the first and second neural networks that SC 37 recites does not render the claim selection inoperative or even ambiguous or unclear. In fact, this is a main goal of SC 37: “[b]y combining multiple algorithms in a way that achieves high *combined frame rate* with higher quality prediction, the system improves outcomes on a limited resource budget.”

EX1001, 21:52-54. Demonstrative 6 shows such an improvement, because either the total processing time would have been longer if only First NN were used, and the quality of predictions would have suffered if only Second NN were used.

165. In summary, for the foregoing reasons, SCs 21-37 are not indefinite.

C. Ground 23 – Beard, Laufer, Pierce, Khan, and Kumar (the “BLPKK combination”) Does Not Render SCs 21-27, 29, 31-33, and 35 Obvious

1. A POSITA Would Not Combine BLPKK

166. In discussing the motivation to combine the above-identified references, Petitioners allege that a POSITA would combine Beard, Laufer, Khan, and Kumar as discussed in the Petition and Dr. Singhose’s first declaration. Opp. at 11-12; EX1039, ¶146. The Petition alleged, in part, that a POSITA would have combined Laufer with Beard (yielding the “BL” combination) because Laufer describes “an automated technique of using machine learning to detect aircraft targets in captured images.” Pet. at 12; EX1003 ¶112. Petitioners added Khan to Beard and Laufer (yielding the “BLKh” combination) alleging, in part, that in “the combination, the detection of targets ... would include identifying a model type of each target.” Pet. at 36; EX1003 ¶152.

167. Separately, Petitioners added Kumar to Beard and Laufer (yielding the “BLKu” combination) contending, in part, because “the distance determination taught by Beard and Laufer” may additionally be based on “a known physical size

of a type of a target.” Pet. at 40; EX1003 ¶159. Petitioners now allege that a POSITA would add Pierce to Beard, Laufer, Khan, and Kumar, yielding the BLPKK combination, because Pierce **allegedly** teaches “a drone having an on-board object tracking module that implements” neural networks.” Opp. at 11-12; EX1039, ¶146. Dr. Singhose argues that if the object tracking module is not on-board, Pierce could be readily modified such that its on-board processor implements the object tracking module’s neural network(s). EX1039, ¶181.

168. In alleging that BL, BLKh, and BLKu combinations were within the scope of a POSITA and that a POSITA would have expected these combinations to succeed, Petitioners make substantially the same arguments with respect to each of Laufer, Khan, and Kumar. In particular, Petitioners or Dr. Singhose state that the “combination simply would have entailed **programming**” Beard’s image processor, and that “machine learning object detection” per Laufer and “object classification” per Khan were well-known and the corresponding algorithms were widely available. EX1003 ¶¶114, 155, 161; Pet. at 13, 37.

169. Regarding BLPKK combination, Petitioners state that Pierce provides “implementation details for how a drone utilizes on-board NNs” and that the combination would have been “applying a known technique.” Opp. at 12; EX1039, ¶180. Petitioners and Dr. Singhose do not cite any evidence showing that the contemplated modifications were within the scope of a POSITA.

170. These assertions are false because some of the modifications that the BLPKK combination requires were reported as “really a complex task,” yielding “terrible” results, and requiring writing of custom scripts, and “a bit of math.” *See supra* § VI.B.1. Therefore, a POSITA would not have found these combinations to succeed, as I explain in details below, identifying several flaws in the proposed BLPKK combination.

a Pierce does not teach or suggest “applying at the reference aerial vehicle a machine learning model”

171. The fundamental flaw in Petitioners’ assertion that Pierce teaches applying a neural network (an ML model) on a drone is that they erroneously conclude that a feature of *some* embodiments (an on-board drone system) universally applies to *all* of the disclosed embodiments. Pierce does not state or even imply this, and in fact, discloses the opposite with respect to embodiments using neural networks. Moreover, a POSITA would have understood that modifying Pierce’s on-board drone system embodiment to include a neural network would have been extremely challenging, if not impossible.

172. As I noted in my Initial Declaration, in the entire disclosure Pierce states once that “In various embodiments, drone system 600 may be drone 102.” EX1005 ¶42; EX2025 ¶198. I also noted that the drone system 600 includes “program instructions for ... object detection by a neural network.” EX2025 ¶198 (quoting EX1005 ¶44). As I further explained in the Initial Declaration, with

reference to Pierce’s FIG. 4, that its “neural network system 500” “operates as a *streaming server*” that a POSITA would understand as not a drone component but as a separate, ground-based server. *Id.* ¶¶200-201, 203 (quoting EX1005 ¶38; EX2020; EX2021).

173. In the Opposition, Petitioners allege that Pierce teaches “a drone having an on-board object tracking module that implements neural networks for object detection.” Opp. at 11-12 (citing EX1005 ¶¶27, 42, 44-46, FIGS. 1, 6); EX1039, ¶181. Pierce does not explicitly disclose “a drone having an on-board object tracking module,” and suggests to the contrary. Specifically, Pierce states that “FIG. 6 illustrates **one example** of a drone system 600.” EX1005 ¶42, FIG. 6 (depicting the object tracking module 615). Thus, FIG. 6 does not represent all drone systems 600, but **one example** of such systems. FIG. 6 does not depict a drone including the system 600, and FIG. 1 does not depict a drone system. *Id.* FIGS. 1, 6.

174. Importantly, and as I discussed in my Initial Declaration, in the very next statement after stating that in “various embodiments, drone system may be drone 102,” Pierce states: “in **other embodiments**, various elements of drone system 600 may correspond to **separate components**, including drone 102, **a server**, a controller, etc.” *Id.* ¶42; EX2025 ¶199. The only description Pierce provides about its “object tracking module 615” is that it may “process a neural

network detection system.” EX1005 ¶45. Therefore, a POSITA would have understood the object tracking module 615 as including the neural network system 500. In such “other embodiments,” the drone 102 and a “server,” such as the “neural network system 500” “that operates as a **streaming server**,” are separate components. Consistent with such other systems, the system of Pierce’s claim 11 even provides “one or more processors and memory separately from the drone. EX1005 at 5 (claim 11).

175. Thus, contrary to Petitioners’ contention that the “object tracking module 615” is a part of a drone, FIG. 6 appears to correspond to such “other embodiments,” where the neural network(s) of the object tracking module 615 are implemented by the neural network system 500 – a separate component operating as a streaming server.

176. Additionally, Pierce also states that “a drone system 600 ... includes ... an **interface 611** ... and may operate as a **streaming server**,” i.e., like the “neural network system 500” “operates as streaming server.” EX1005 ¶¶38, ¶42; EX2025 ¶199. Pierce’s disclosure of the drone system 600 operating as a streaming server and its “neural network system 500” that “operates as a **streaming server**” is substantially identical, as a side-by-side comparison below shows:

Drone System 600	Neural Network System 500
“According to particular embodiments, a drone system 600, suitable for implementing particular embodiments of the present disclosure, includes a processor 601, memory 603, an interface 611, and a bus 615 (e.g., a PCI bus or other interconnection fabric) and may operate as a streaming server.” EX1005 ¶42.	“FIG. 5 illustrates one example of a neural network system 500, in accordance with one or more embodiments. According to particular embodiments, a system 500, suitable for implementing particular embodiments of the present disclosure, includes a processor 501, memory 503, an interface 511, and a bus 515 (e.g., a PCI bus or other interconnection fabric) and operates as a streaming server.” <i>Id.</i> ¶38.
“The interface 611 is typically configured to send and receive data packets or data segments over a network. Particular examples of interfaces supports include Ethernet interfaces, frame relay interfaces, cable interfaces, DSL interfaces, token ring interfaces, and the like.” <i>Id.</i> ¶¶42-43.	“The interface 511 is typically configured to send and receive data packets or data segments over a network. Particular examples of interfaces supports include Ethernet interfaces, frame relay interfaces, cable interfaces, DSL interfaces, token ring interfaces, and the like.” <i>Id.</i> ¶¶38-39.

177. In light of this substantially identical description of some embodiments of the drone system 600 and the neural network system 500, a POSITA would have understood that the drone system 600 that includes the object

tracking module 615, and thus its neural network(s), is a ground-based system that includes a drone and operates as a “streaming server,” sending and receiving “segments of data over a network” using a wired interfaces, such as “**Ethernet interfaces**, frame relay interfaces, **cable interfaces**, **DSL interfaces**,” etc.

178. Therefore, a POSITA would have understood a drone system implementing neural networks as part of the object tracking module 615 as a ground-based system that is not and cannot be on-board Pierce’s UAV. Accordingly, Pierce does not teach or suggest “applying at the reference aerial vehicle a machine learning model,” such as a neural network. Moreover, as explained below, modifying the drone system 600 that is a drone to further include a neural network system would have been extremely challenging, if not impossible. Neither Pierce nor any other prior art in the BLPKK combination provides any guidance whatsoever for such a modification.

b A POSITA would have understood that Pierce’s on-board drone system could not simply be programmed to implement Pierce’s neural networks

179. To the extent Petitioners allege that Pierce’s drone system that is a drone could be readily modified to implement Pierce’s neural network on-board, and/or to include on-board the object tracking module 615 of a different, separate-component embodiment, *see* EX1039, ¶181, a POSITA would have known the opposite to be true.

180. **The control loop problem:** Around 2017, the excitement around deep neural networks for object detection (like YOLO and SSD) led many researchers to focus heavily on the perception problem “can we detect and classify a target?” Many researchers recognized, however, the far harder control loop or feedback problem: even if a neural network would perfectly identify a target (e.g., a target drone) in a camera frame, that detection must be converted into real-world three-dimensional position and velocity estimates, which requires solving depth ambiguity from a monocular camera, accounting for the interceptor drone’s own ego-motion.

181. From there, a controller must generate thrust and attitude commands fast enough to track a potentially agile, unpredictably maneuvering target. During this, the camera’s field of view may shift with every adjustment to the drone movement, potentially losing the target entirely. The latency of running a deep network without a server system (a real constraint in 2017) would have further degraded the control bandwidth, making tight pursuit loops nearly impossible.

182. As such, many researchers at the time understood that the on-board execution of a neural network for detection, though beneficial in some ways, would conflate solving the detection problem with solving the target following and/or interception problems, ignoring the cascaded estimation, latency, and feedback-stability challenges that intervene detecting a target and actually tracking

or catching it. Therefore, if neural networks were used for object detection in the images / video captured by a drone, they were implemented using ground-based powerful servers, having access to multi-processor, parallelized-computing systems.

183. I am informed by Counsel and understand that even when “combining multiple embodiments from a single reference ... there must be a motivation to make the combination and a reasonable expectation that such a combination would be successful.” *In re Stepan Co.*, 868 F. 3d 1342, 1346 n.1 (Fed. Cir. 2017). A POSITA would not have been motivated to configure Pierce’s on-board processor (if any) to implement neural network(s).

184. Pierce discloses that the drone system 600 includes “a processor 601” that using “appropriate software or firmware,” can process “inputs through various computational layers and algorithms.” EX1005 ¶42. But Pierce is silent on the type of the processor. *See generally id.* Pierce also describes the use of neural networks for image analysis, but does not identify any particular kind of neural networks. *See generally id.*

185. I explained above that before 2017 (and even today) a POSITA would have known that a neural network is a computationally intensive process and that neural networks are typically implemented using a GPU. *See supra* § VI.B.1; EX1036 at 48 (p. 779) (disclosing “Titan X GPU implementation”); EX2039,

Abstract (“GPUs are routinely used instead [of CPUs] to train and run [neural] networks”); EX2043 at 552 (machine vision techniques “mostly need vast quantity of GPU power” and “non-GPU computer[]” implementations of “YOLO” and “Faster R-CNN” are “unsuitable for real-time application”). As such, neural network parameters (e.g., weights) were often available for frameworks used for GPU execution not for the frameworks used by a non-GPU processor. *See* EX2044 at 7-10 (describing that weights were readily available for Darknet but not for Keras and Metal).

186. Thus, a POSITA would have known that to execute a particular neural network (Pierce’s or any other) using a generic processor, such as a CPU, the processor must be specifically configured for such an execution. A POSITA would have also known that such configuration requires a substantial effort along at least three dimensions. In this **three-step reconfiguration**, first, the CPU must be provided with streaming SIMD extensions (SSE). EX2041 at 1 (p. 47). Second, a framework suitable for the execution of a particular neural network must be selected from choices including TensorFlow, Darknet, Darkflow, OpenCV, Keras, Metal, etc. *See* EX2043 at 2 (p. 553); EX2044 at 6-7.

187. Third, while a neural network algorithm may be well-known and may be widely available, it would have to be deployed in a framework that is suitable for CPU / image processor operation for the proposed combination to function. In

that case, the parameters of the algorithm that may be available for a different framework (e.g., one suitable for a GPU), must be recomputed for the framework selected for CPU / image processor application. This task involves writing of custom scripts and deriving and using the required math equations. *See* EX2044 at 7-10.

188. These tasks, especially the latter two, are not simple tasks and were not within the scope of a POSITA. For the second task there are several different (at least six) frameworks available, *see* EX2043 at 2 (p. 553); EX2044 at 6-7, and a POSITA would have had to evaluate each one to determine the performance of the neural network to be implemented on a CPU. The performance itself has two dimensions: accuracy and speed. While one framework may yield a better accuracy, the other one may yield a better execution time. Some frameworks may be found not suitable at all. *See* EX2043 at 553 (“DarkFlow installation in windows is **really a complex task**” and upon completing it, “the result [of running YOLOv3] was **terrible**”). Also, before a candidate framework is tested, parameters for that framework need to be computed. *See* EX2044 at 7-10 (describing converting Darknet “weights” for tiny YOLO into Keras weights using an available “YAD2K” script, and then converting Keras weights into Metal weights using a custom script and “a bit of math,” involving specific mathematical formulae). The parameters

may also need to be tweaked further by trial and error in several rounds of testing, to attain a desired accuracy.

189. A POSITA thus would have known that choosing a neural network algorithm (even when known and widely distributed) and implementing it on a generic processor is a challenging **three-step reconfiguration** task, with the likelihood of many experiments failing. A POSITA would have also known that recomputing parameters accurately requires advanced knowledge of neural networks well beyond the scope of a POSITA.

190. Petitioners and Dr. Singhose do not explain how a POSITA would have addressed these substantial issues. They do not even acknowledge the issues, and simply describe the implementation on a generic processor as “applying a known technique.” While the neural network algorithms are known, how to port a neural network to a generic processor and a different framework were not generally known. Therefore, a POSITA would have expected the opposite of Petitioners’ contentions. *See* EX2043 at 553 (“DarkFlow installation in windows is **really a complex task**” and upon completing it, “the result [of running YOLOv3] was **terrible**”). A POSITA would have expected an attempt to implement a chosen neural network (Pierce’s or any other) on Pierce’s processor outside of its neural network system 500 extremely difficult, if feasible at all.

c A POSITA would have understood that Beard's image processor could not simply be programmed according to Laufer and/or Pierce

191. According to Petitioners, a POSITA would have combined Beard, Laufer, and Pierce because Laufer and Pierce would have allegedly improved Beard's system by incorporating Laufer's machine learning systems and Pierce's neural networks. *See supra* ¶¶166-170; Pet. at 12; EX1003 ¶112; Opp. at 12; EX1039, ¶146. The examples of machine learning that Laufer describes are all neural networks. EX1010 ¶31. Petitioners or Dr. Singhose also state that the "combination simply would have entailed **programming**" Beard's image processor, that Laufer's and Pierce's neural networks were well-known and widely available, and the required modification would have been "applying a known technique." *See supra* ¶¶166-170; EX1003 ¶¶114, 155, 161; Pet. at 13, 37; Opp. at 12; EX1033 ¶146.

192. Thus, Petitioners allege that implementing Laufer's and Pierce's neural networks on Beard's drone would have required nothing more than programming Beard's image processor using known, available algorithms. This couldn't be farther from the truth. Beard does not provide any specifics about its image processor, *see generally* EX1009, and as such, a POSITA would have understood it as a generic processor / CPU or a digital signal processor (DSP), but not a GPU. Therefore, a POSITA would have readily understood that the **three-**

step reconfiguration process needed for Pierce’s processor would also be necessary for Beard’s image processor. In fact, in this case, the reconfiguration process would need to be performed for Laufer’s neural network and then once again for Pierce’s neural network. *See supra* VIII.C.1.b.

193. For the BLPKK combination, Petitioners and Dr. Singhose do not even acknowledge and do not explain how a POSITA would have addressed these substantial processor reconfiguration issues. Therefore, a POSITA would not have expected an attempt to implement Laufer’s and Pierce’s neural network on Beard’s image processor to succeed, finding such an endeavor extremely difficult, if feasible at all.

d The BLPKK combination is an amalgam of duplicative and incompatible features

194. The BLPKK combination leads to an amalgam of features that are duplicative and/or incompatible.

195. Duplicate Classifiers: First, Laufer discloses a neural-network-based classifier that can “classify whether a target is a threat or not.” EX101, ¶31. Khan’s classifier is not a neural network, however; it is a support vector machine (SVM) that can determine a “specific vehicle-type.” EX1021, ¶¶14, 62. While they both may be used for image classification, a deep neural network used for image analysis and an SVM are quite different. An SVM requires “feature” extraction, e.g., using histograms of oriented gradients, while a deep neural network used for

image analysis performs feature extraction automatically, in one or more hidden layers. *See infra* ¶¶198-200. Thus, Laufer and Khan disclose classifiers that have different structures and operate differently.

196. Yet, Petitioners do not explain **how** a POSITA would have implemented these two different techniques on Beard's image processor. For instance, whether Beard's processor would have been dual configured to implement both an SVM and a neural network or, in the alternative, whether a POSITA would have modified either Laufer's neural network (for determining if an object is a drone, and if it is a threat) to provide Khan's functionality (model/type classification), or vice versa, and then implemented the modified neural network or modified SVM on Beard's processor. Petitioners also do not explain **how** a POSITA would have performed such a dual configuration or modification and **why** a POSITA would expect such a complex change to succeed.

197. Regarding the alternative noted above, modifying a first neural network to perform the function of a second neural network is not a simple task. One cannot simply insert the model of the second neural network into the first neural network. Instead, the first neural network must be retrained entirely, to perform both its original function and the second neural network's function. This can adversely impact the accuracy of either or both functions and/or the speed of computation of either or both functions. Petitioners do not discuss why it would

have been within the scope of a POSITA to undertake such a complex modification.

198. Duplicate Feature Extraction: Before explaining this flaw, I note that “feature” in the context of using machine learning for image-based object detection or classification is a **characteristic of the image** that is beneficial in detecting / classifying the object. It is not a feature of the object to be detected. For example, in this context, if the object is human, the features are not head, arms, legs, or if the object is a drone, the features are not rotors, cockpit, landing gear, etc. *See* EX2042 at 2, Fig. 2 (depicting HOG features). As such, feature extraction in this context is not the same as or even related to the “detected features” that SC 21 recites.

199. Laufer, Pierce, and Kumar all describe using a neural network for target detection. EX1010 ¶ 31; EX1005 ¶¶38, 45, FIGS. 5, 6; EX1008, ¶35, FIG. 4. In a deep neural network used for image analysis, features are learned automatically, by one or more hidden layers of the neural network itself. In fact, eliminating the need for the detection of handcrafted features (e.g., HoG features used by an SVM classifier) is considered one of the advantages of deep neural networks. *See* EX2042 at 1-2 (“learned features” are learned “directly from data using deep convolutional neural networks,” “are gaining popularity, as they are outperforming hand-crafted features in many [computer vision] tasks,” and that an SVM classifier detects objects using handcrafted HoG features).

200. Khan in contrast, uses an SVM classifier that requires handcrafted features. Khan uses the Histogram of Oriented Gradients (HoG) technique for feature extraction. EX1021, ¶49. The HoG features of a model are matched to those of an image, and p and n match scores are computed. *Id.* ¶¶45, 49, 54. The classification performed by Khan’s SVM uses these computed p and n match scores to classify the detected image. *Id.* ¶62 (“the detected vehicle 702 in the captured image 700 is classified using a two stage classifier method based on the derived p-scores and n-scores”), FIG. 11. Thus, Khan’s classification cannot be performed without performing HoG-based feature extraction and computation of the match scores based on the extracted features.

201. Because Laufer, Pierce, and Kumar’s neural networks “learn” features in a hidden layer, those features are not output by the neural network. Moreover, none of Laufer, Pierce, and Kumar discloses any technique whatsoever that can produce the match scores that Khan requires. Beard is silent on extracted features and associated match scores. Thus, in the proposed combination, even though the neural networks of Laufer, Pierce, and Kumar learn features, Khan’s additional HoG-based feature extraction would still also be required. Petitioners and Dr. Singhose do not explain, why a POSITA would employ two different techniques for the same task, introducing a redundancy and computational burden in the system.

202. Furthermore, the problem associated with implementing Laufer's and Pierce's neural networks using Beard's image processor compounds further with the addition of Khan. Now, Beard's image processor would need to: (i) implement Laufer's neural network, Pierce's neural network, and Khan's classifier, each separately; or (ii) implement a modified neural network performing Laufer's and Pierce's functions, and additionally implement Khan's classifier. Per both alternatives, Beard's image processor would also need to implement Khan's HoG algorithm. Petitioners do not explain which of these alternatives (or any other alternative) would be implemented, and how a POSITA would perform the required modifications, and account for the execution of duplicative feature extraction tasks. Petitioners also do not explain why a POSITA would have expected the combination to succeed by overcoming the significant challenges a POSITA would face.

e The BLPKK combination renders Beard inoperable for its intended purpose

203. I am informed by Counsel and understand that if "references taken in combination would produce a 'seemingly inoperative device,' ... such references teach away from the combination and thus cannot serve as predicates for a prima facie case of obviousness." *McGinley v. Franklin Sports, Inc.*, 262 F. 3d 1339, 1354 (Fed. Cir. 2001) (citing *In re Gordon*, 733 F.2d 900, 902, 221 (Fed. Cir. 1984).

204. The BLPKK combination would likely render Beard inoperable for its intended purpose. Beard describes “capture of **real-time streaming video** from multiple unique perspectives, e.g., providing a skier’s eye view of the downhill run.” EX1009, 3:30-35; *see id.* 1:56-59. To this end, Beard’s

action camera system 200 uses information about the current orientation of the target 118 to the frame 110 to adjust the speed or position of UAV 100 and maintain the initial orientation (e.g., **keep a skier centrally framed at a fixed distance** as s/he proceeds through a downhill run).

Id. 3:26-30.

205. This requires that Beard perform its image processing in real time. It was well understood in 2017, that if a neural network is implemented by a generic processor, the neural network would not be executed in real-time. EX2045 at 1-2 (describing YOLO delivers high accuracy at high speed using a GPU, but “cannot work in constrained environments”); EX2051 at 1-2 (stating that YOLO performs well “on high-end desktop GPUs” but is “too large for many edge and mobile scenarios” such as on-board a drone); EX2052 at 3 (“Tiny-yolo was important because it [and not YOLO could yield] reasonable results when deployed to the limited hardwar[e] of a mobile device”); EX2044 at 6-7 (the “full YOLOv2 network has three times as many layers and is a bit too big to run fast enough on current [i.e., 2017] iPhones”).

206. As such, configuring Beard's image processor to implement Laufer's and Pierce's neural networks⁷ and additionally Khan's HoG algorithm and SVM, would prevent that processor from performing real-time image analysis, not solving the control loop challenge that I discussed above. *See supra* ¶¶180-182. This would render Beard's system inoperable for its intended purpose.

207. In summary, for the BLPKK combination:

- (i) a POSITA would have reasonably expected the implementation of any of the neural networks in the cited prior art by Beard's image processor to **fail** (if feasible at all) – not succeed;
- (ii) notwithstanding that difficulty, a POSITA would have understood that Pierce's neural network, Khan's SVM, and the HoG technique were all necessary for a proper functioning of Pierce's and Khan's techniques, and would have reasonably expected that:
 - a) such a complex implementation would not succeed; and

⁷ I note that SC 21 does not recite a neural network and only recites a machine learning model. But, the only machine learning models that Laufer, Pierce, and Kumar disclose are neural networks. Therefore, the BLPKK combination requires executing neural networks, and also an SVM.

- b) if successful at all, the implementation of these three computation-intensive processes would render Beard's system inoperable for its intended purpose.

208. As such, a POSITA would not have even attempted to combine BLPKK.

2. The BLPKK Combinations Does Not Teach or Suggest Each and Every Limitation of PSC 21

a Claim element [21.b]: "applying at the reference aerial vehicle a machine learning model"

209. In connection with the above-identified claim element, Beard is silent on a machine learning model. Therefore, Beard does not teach the above-recited claim element.

210. Petitioners do not allege that Laufer's machine learning model is applied on-board an RAV, and Laufer also does not teach this claim element. Laufer's machine learning model is implemented on a ground-based central controller. Specifically, Laufer discloses that the "[d]rone identification module 220 permits central controller 115 to positively detect target as threats." EX1010 ¶31. "Machine learning algorithms, for example Deep Convolution Neural Network" may be used for object and drone classification. *Id.* ¶31, FIG. 2 (depicting the drone identification module 220 as part of the central controller 115). The central controller 115 has wired connections to other ground-based

components, and is thus ground based. *See* EX2025 ¶91. Thus, Laufer’s neural network is implemented in the ground-based central controller and not on-board an RAV.

211. Petitioners also do not allege that Kumar’s processor is on-board an RAV, and Kumar does not teach this claim element either. Kumar’s “neural network system 400” includes a processor 401. EX1007 ¶40; EX1008 ¶35, FIG. 4. The neural network system 400 includes wired interfaces such as “Ethernet,” “cable,” and “DSL.” EX1007 ¶¶40-41; EX1008 ¶¶35-36. As such, Kumar’s network system 400 is also ground-based and not on-board an RAV. *See* EX2025 ¶204.

212. Khan discloses an “on-board computing platform 114.” EX1021 ¶39, FIG. 2. Khan discloses an SVM, an ML model, though not a neural network. EX1021 ¶¶14, 16, 82; *see generally id.* Khan also states that its computing platform 114 may include “a work-station ... comprising one or more processors” or “may be implemented as or part of and integrated circuit, such as a” GPU, DSP “implemented in an FPGA or [an application specific integrated circuit] ASIC.” *Id.* Petitioners and Dr. Singhose have not alleged that Beard’s image processor may be replaced and enhanced according to Khan’s teachings.

213. Such a modification is not straightforward because replacing a generic processor with a GPU or adding a GPU to the processor board requires redesigning

the processor board. It also requires providing memory and clock circuitry suitable for use by GPU, and programming the GPU to implement Beard's functions such as identifying bounding boxes. Although Petitioners allege that Khan discloses identifying bounding boxes, that is not true, as I explain below. These modifications are complex and would not have been within the scope of a POSITA. Regardless, Petitioners do not discuss such a modification.

214. Regarding Pierce, Petitioners appear to allege that because Pierce states that in "various embodiments, drone system 600 may be drone 102," and that "one example" drone system depicts an "object tracking module 615" (that implements a neural network), any drone system that is on-board a drone necessarily includes the "object tracking module." I have explained above that this is erroneous and is inconsistent with the description of Pierce's neural network system 500 that may be implemented within the object tracking module. *See supra* §§ VIII.C.1.a, VIII.C.1.b. Thus, Pierce also does not teach or suggest this claim element.

215. Petitioners also appear to allege that Beard's image processor can implement Laufer and Pierce's neural networks and Khan's SVM classifier. A POSITA would not have known how to implement neural networks on Beard's generic image processor, and would have known this to be a trial-and-error based rather difficult and tenuous task, if not an impossible one. *See supra* §§ VIII.C.1.b,

VIII.C.1.c. Therefore, the BLPKK combination does not teach or suggest “applying at the reference aerial vehicle a machine learning model.”

b [21.b] “applying ... a machine learning model to the image ... to identify a bounding box outlining a portion of the image that includes detected features of the target”

216. Petitioners cite Khan and Kumar as allegedly disclosing this limitation. Opp. at 12-15; EX1039, ¶¶149-155. In particular, Petitioners state that Khan’s “computer processes captured images 700 by determining “salient features of a vehicle 702” in images; the features are “shape and appearance” features at feature locations 804, 806, and 808, located within a bounding box surrounding vehicle 702 shown by Figs. 7 and 8.” Opp. at 13; EX1039, ¶151.

217. These arguments are wrong for at least two reasons. First, the bounding box seen only in FIG. 7 is **provided by the illustrator**, to depict a related zoomed-in view labeled “Coupe to F-Sedan.” EX1021, FIG. 7. That illustrator-drawn box is not identified by any of process that Khan describes, including the HoG algorithm and the SVM.

218. Second, to the extent any box that may be **inferred** by joining the salient feature locations 804, 806, 808 associated with the detected vehicle 702, that box is not identified by Khan’s SVM, i.e., the alleged ML model, because the SVM does not determine the locations 804, 806, 808. Rather, those locations are determined using certain traditional image processing algorithms.

219. In particular, Khan discloses “representative collection of images 600 of nine predetermined vehicle models 602.” EX1021, ¶45, FIG. 6. Khan further states that the “salient locations 604, 606, 608 have been **empirically** determined.” *Id.* Khan also states that the “salient locations 604, 606, 608 may be marked either *a-priori* by a user or obtained automatically using either **mesh segmentation** methods ... or **an analysis of maximas/minimas in the surface curvature.**” *Id.* ¶46. Khan further states that “for each of the salient feature locations 604, 606, 608, **Histogram of Oriented Gradients (HoG) features 802, 804** are obtained for both the rendered view of the 3D vehicle models 602 as well as the captured image 700, respectively.” *Id.* ¶49.

220. Thus, the salient feature locations 804, 806, 808 in the captured image 700 correspond to the empirically determined locations 606, 606, 608 in a predetermined model. The features at these locations are determined using the HoG algorithm and, as such, Khan refers to them as “HoG features.” *Id.* Khan further states: “The HoG features 802, 804 are formed into feature **vectors**. These feature vectors are then matched as described hereinbelow in connections with steps S4-S9 and FIGS. 9, and 10.” *Id.* Step S9 describes the use of the SVM classifier. *See id.* ¶62.

221. Thus, Khan’s SVM classifier, i.e., the alleged ML model, does not generate the HoG features and does not identify their locations. Rather, techniques

such as mesh segmentation or analysis of maximas/minimas in the surface curvature together with the HoG algorithm determine certain locations and features at those locations. These techniques are classic image processing techniques; they are not ML models.

222. Mesh segmentation involve cutting a “mesh by edges of triangles using partitioning measures.” EX2055 at 1. Analysis of maximas/minimas in the surface curvature also involves analyzing “surfaces approximated by dense triangle meshes.” EX2056, Abstract. In particular, the technique uses “curvature statistics in order to ignore insignificant curvature extrema” and adapts “an image processing technique for triangle meshes ... for bridging gaps between connected chains of curvature extrema triangles.” Id. at 2.

223. The HoG algorithm involves histogram computations of pixels in an image. *See* EX2057 at 3 (describing HoG as dividing an image into cells, “accumulating a local 1-D histogram of gradient directions or edge orientations over the pixels of the cell,” and using a “measure of local histogram ‘energy’ over ... blocks ... to normalize all of the cells in the block”).

224. In summary, the bounding box to which Petitioners refer is provided by the illustrator and not any of Khan’s processing techniques. Moreover, to the extent any bounding box may be **inferred** from Khan’s salient feature locations 804, 806, 808 that are associated with the detected vehicle 702, those locations are

determined using traditional image processing algorithms such mesh segmentation, analysis of maximas/minimas in the surface curvature, and the HoG algorithm. The inferred box is not identified by Khan’s SVM, i.e., the alleged ML model. As such, Khan does not teach or even suggest “applying ... a machine learning model to the image ... to identify a bounding box outlining a portion of the image that includes detected features of the target.”

225. For its part, Kumar discloses minimal bounding boxes. EX1007, Abstract, ¶¶17-19, FIGS. 3A. But Kumar does not describe “detected features of a target.” *See generally, id.* I note that Kumar references U.S. Provisional Application No. 62/261,260 (the “’260 Prov.”) as describing “a neural network” outputting the bounding boxes that Kumar describes. *Id.* ¶19. Petitioners do not cite the ’260 Prov. *See generally* Opp.; EX1039. Nevertheless, the ’260 Prov. also does not disclose a neural network identifying a bounding box outlining “detected features of a target.” The ’260 Prov. also mentions minimal bounding boxes but does not describe how such bounding boxes are generated. *See generally, EX2107.*

226. Beard, Laufer, and Pierce do not disclose this feature, and Petitioners do not allege that any of them does. Therefore, Khan and Kumar, individually or in the BLPKK combination, do not teach or suggest “applying ... a machine learning model to the image ... to identify a bounding box outlining a portion of the image that includes detected features of the target,” as SC 21 recites.

227. In conclusion in connection with SC 21, a POSITA would not have combined BLPKK. Regardless, the combination does not teach or suggest all of the limitations of SC 21. Therefore, in my opinion, SC 21 is patentable over the BLPKK combination.

D. Grounds 24-26 Do Not Render SCs 28, 30, and 34 Obvious

228. Grounds 24-26 challenge validity of dependent claims 28, 30, and 34, respectively, adding Malecki, Sahasrabudhe, and Hogasten, respectively, to the BLPKK combination. Opp. at 17; EX1039, ¶¶165-168. These claims depend from SC 21 and are patentable at least because SC 21 is patentable, as discussed above. *See supra* § VIII.C.

E. Ground 27 – Beard, Laufer, Pierce, and Ning (the BLPN Combination”) Does Not Render SC 36 Obvious

1. A POSITA Would Not Combine BLPN

229. In discussing the motivation to combine the above-identified references, Petitioners allege that a POSITA would combine Beard, Laufer, and Pierce as discussed for SC 21. Opp. at 17-18; EX1039, ¶¶169-170. In the context of SC 21, Petitioners alleged, in part, that a POSITA would have combined Laufer with Beard (yielding the “BL” combination) because Laufer describes “an automated technique of using machine learning to detect aircraft targets in captured images.” Pet. at 12; EX1003 ¶112. Petitioners then allege for SC 21 that a POSITA would add Pierce to Beard and Laufer because Pierce **allegedly** teaches “a drone

having an on-board object tracking module that implements neural networks.”

Opp. at 11-12; EX1039, ¶¶146, 181.

230. Regarding the further addition of Ning, Petitioners and Dr. Singhose allege that “the need for fast and accurate object detection for vehicle control was known” and that “Ning’s YOLO and LSTM” can “enhanc[e] Beard’s object detection and tracking.” Opp. at 17-18; EX1039, ¶¶169-170.

231. In alleging that the BL combination was within the scope of a POSITA and that a POSITA would have expected these combinations to succeed, Petitioners and Dr. Singhose state that the “combination would have simply entailed **programming**” Beard’s image processor, and that “machine learning object detection” was well-known and that the corresponding algorithms were widely available. EX1003 ¶¶114; Pet. at 13. Regarding adding Pierce to the combination, Petitioners state that it provides “implementation details for how a drone utilizes on-board neural networks” and that the combination would have been “applying a known technique.” Opp. at 12; EX1039, ¶146.

232. In alleging that Ning’s addition to the combination was also within the scope of a POSITA, Petitioners allege that “using YOLO was a known favorable solution applicable to object recognition and tracking algorithms used by drones.” Opp. at 18; EX1039, ¶169. Petitioners and Dr. Singhose also quote Ning as stating that its “system ‘can be amounted on any suitable vehicle,” and argue that Ning’s

system “is thus compatible with the Beard-Laufer-Pierce drone system.” Opp. at 18; EX1039, ¶169.

233. These assertions are false. As a threshold matter, a POSITA would have recognized that while Ning’s system employs a YOLO and an LSTM, *see* Opp. at 19-20; EX1039, ¶175-178; EX1034, 2:15-19, Ning’s LSTM simply could not be and could not have been configured as the claimed “second neural network,” due to an inherent drawback of LSTMs. *See id.* ¶¶274-283. Moreover, some of the modifications that the BLPN combination requires were reported in the literature by 2017 as “really a complex task,” yielding “terrible” results.” *See supra* § VI.B.1. Even the aforementioned Ning statement quoted by Petitioners and Dr. Singhose in support of the combination is misleading; it omits key limitations Ning himself associated with the quoted statement. In addition, as explained in detail below, there are several additional flaws in the proposed BLPN combination. Therefore, a POSITA would not have had any expectation of success in achieving the claimed invention by combining Ning with the other references.

a Pierce does not teach or suggest “applying at the reference aerial vehicle a machine learning model”

234. A major flaw in the BLPKK combination also arises in the BLPN combination. *See supra* § VIII.C.1.a.

b A POSITA would have understood that Pierce's on-board drone system could not simply be programmed to implement Pierce's neural networks

235. It does not appear that in the BLPN combination Petitioners rely on Pierce's neural network. They appear to rely only on Laufer's and Ning's neural network. Nevertheless, to the extent the use of Pierce's neural network is contemplated or required in the BLPN combination, the above-identified flaw in this combination is the same as that in the BLPKK combination, creating the control loop problem that I discussed above. *See supra* § VIII.C.1.b.

c A POSITA would have understood that Beard's image processor could not simply be programmed according to Laufer and Ning

236. According to Petitioners, combining Beard and Laufer was allegedly within the scope of a POSITA because the “combination simply would have entailed **programming**” Beard's image processor, that Laufer's neural networks were well-known and widely available, and the required modification would have been “applying a known technique.” *See supra* ¶¶229-231; EX1003 ¶114; Pet. at 13; EX1039, ¶179.

237. In alleging that Ning's addition was within the scope of a POSITA, Petitioners allege that “using YOLO was a known favorable solution applicable to object recognition and tracking algorithms used by drones.” Opp. at 18; EX1039, ¶169. Petitioners and Dr. Singhose also quote Ning as stating that its “system ‘can

be []mounted on any suitable vehicle,” and argue that Ning’s system “is thus compatible with the Beard-Laufer-Pierce drone system.” Opp. at 18; EX1039, ¶169.

238. Thus, Petitioners allege that implementing Laufer’s and/or Ning’s neural networks on Beard’s drone would have required nothing more than programming Beard’s image processor using known, available algorithms, and because Ning allegedly taught that the YOLO was suited for any vehicle. This is not true.

239. Beard, which teaches a drone-hosted computer system, does not provide any specifics about its image processor, *see generally* EX1009, and as such, a POSITA would have understood it as a generic processor or a digital signal processor (DSP), but not a GPU. Therefore, a POSITA would have readily understood that the **three-step reconfiguration process** that was necessary for the BLPKK combination would also be necessary for the BLPN combination. In the case of BLPN, the three-step reconfiguration process would be for Laufer’s neural network and/or for each of Ning’s two neural networks. *See supra* VIII.C.1.b. Petitioners and Dr. Singhose however, do not even acknowledge that a POSITA would have need to implement this substantial 3-step processor reconfiguration, let alone explain one would do so (Petitioners and Dr. Singhose, by the way, also failed to provide the required explanation for the BLPKK combination).

240. Moreover, a POSITA would have found implementing Ning’s YOLO on a generic processor on-board a drone extremely challenging, because it was already reported as a practically infeasible task. *See* 1-2 at 1935-36 (YOLO “cannot work in constrained environments”); EX2051 at 1-2 (YOLO is “too large for many edge and mobile scenarios” such as on-board a drone); EX2052 at 3 (Tiny-yolo and not YOLO could yield reasonable results using limited hardware); EX2044 at 6-7 (the “full YOLOv2 ... is a bit too big to run fast enough on current [i.e., 2017] iPhones”).

241. In alleging that Ning’s YOLO is the claimed “first neural network,” Petitioner states: the “’711 Patent says YOLO neural networks have a ‘high robustness’ and ‘high cost,’ and Ning discloses a YOLO neural network.” Petitioner however misquotes the ’711 patent in making this argument, and then draws a misleading and incorrect analogy. The ’711 patent actually states: “[h]igh cost (low refresh rate) – high robustness can include modified **tiny** You only look once (YOLO).” EX2019 ¶150; EX1001, 28:40-47. The word “tiny” in tiny YOLO is not a generic descriptor; it is a specific type of YOLO. *See* EX2045 at 1-2 (pp. 1935-36) (distinguishing tiny YOLO from YOLO); EX2051 at 1-2 (same); EX2052 at 2 (same). A POSITA would also have known that Ning’s traditional YOLO is substantially different, e.g., in terms of size and computation power requirements, from the tiny YOLO described in the ’711 patent.

242. For example, Fang (EX2045) states:

By running on the **powerful GPU platform**, YOLO and its improvements have reached high accuracy and fast speed. However, the **model sizes** of these object detection algorithms **are too large** for constrained environments with limited storage memory devices and they **cannot work** in constrained environments with real-time performance. **Tiny-YOLO-V3** is the latest improvement of YOLO with a **relatively small model size** for constrained environments.

EX2045 at 1-2 (pp. 1935-36) (internal citation omitted).

243. Fang also discloses that the model size of YOLO-v1 and v2 are over 1 GB and 200 MB, respectively, while the model sizes of Tiny-YOLO-v2 and v3 are less than 65 MB. *Id.* at 1942. Fang also describes that billion floating point operations per second, i.e., “BFLOP/s is often used as a measure of the computational power required by a model. The larger the BFLOP/s, the higher the requirements for the device.” *Id.* at 1941. In terms of computing power requirements, Fang further discloses that YOLO-v1 and v2 require over 34 BFLOP/s, while Tiny-YOLO-v2 and v3 require less than 7 BFLOP/s. *Id.* at 1942.

244. Likewise, Wong (EX2051) states:

A particularly interesting family of object detection networks designed around efficiency is the YOLO family of neural network architectures ... which can achieve embedded object detection performance on high-end **desktop GPUs**. However,

these **network architectures remain too large** for many edge and **mobile scenarios** (e.g., ~240MB in the case of the YOLOv3 architecture), and their **inference speeds drop considerably when running on** edge and **mobile processors** due to computational complexity (e.g., >65B operations in the case of YOLOv3). To address this issue, Redmon *et al.* introduced the **Tiny YOLO family** of network architectures, which has **greatly reduced model sizes** at a cost of object detection performance.

EX2051 at 1-2.

245. Similarly, Apte (EX2052) states:

We focused specifically on modifying and deploying tiny-YOLO, a version of YOLO that uses **9 convolution layers and 6 pooling layers instead of the 24 convolution layers** in the traditional YOLO net.

EX2052 at 2 (internal citation omitted).

246. Apte further states:

For our project we focused on models specific for the VOC task, meaning the number of classes for prediction was 20.

Since there are [] various architectures that can be implemented for this task, the authors of YOLO described a full YOLO version which achieved optimal accuracy and a more compact YOLO called tiny-yolo that run faster but isn't as accurate. **Tiny-yolo was**

important to our project because it allowed us to get reasonable results when deployed to the **limited hardwar[e] of a mobile device**.

Id. at 3.

247. Thus, Fang, Wong, and Apte each describes that tiny YOLO has a different structure than YOLO; the model size and computing power requirements of tiny YOLO are significantly smaller, respectively, than those of YOLO.

Additionally, Apte explains that YOLO includes 24 convolutional layers while tiny YOLO includes only 9, which in part explains the significantly smaller size and computational power requirements of the latter.

248. As another example , Redmon discloses a traditional YOLO (also called a base network) and a fast YOLO. EX1036, Abstract, at 49 (p. 780) (stating that YOLO “network has 24 convolutional layers” and that “Fast YOLO uses a neural network with fewer convolutional layers (9 instead of 24)”). Redmon further states: “[o]ur base network runs at 45 frames per second with no batch processing on a **Titan X GPU** and a fast version runs at more than 150 fps.” *Id.* at 48 (p. 779). A POSITA would have known that a Titan X GPU is a highly specialized, multi-core processor that includes 3,072 CUDA cores and 12 GB memory. EX2058. CUDA is a specialized “parallel computing platform and programming model.” EX2059. Even on a specialized processor such as Titan X, traditional YOLO runs more than 3-times slower than the Fast YOLO (which, like the Tiny YOLO, also has 9 convolution layers).

249. Accordingly, a POSITA would have understood that the traditional YOLO and tiny YOLO are significantly different, and not comparable in terms of structure, size, computation cost, and execution speed. Petitioner provides no information about Ning's YOLO in terms of its computation cost, and simply relies on a falsely-premised comparison between the '711 patent's disclosure and Ning's YOLO. Opp. at 19 (falsely stating that the "'711 Patent says YOLO NNs have a 'high robustness' and 'high cost,' when in fact, the '711 patent describes "modified tiny" YOLO, which is not at all the same as YOLO).

250. Although YOLO may have higher robustness, higher cost, and lower refresh rate than those of tiny YOLO, respectively, (and may thus be alleged to meet the cost requirements of the claimed first neural network), as described above, a POSITA would have understood that YOLO simply cannot be provided on-board Beard's image processor.

251. Finally, Petitioners allege that "Ning's system 'can be []mounted on any suitable vehicle' and is thus compatible with Beard-Laufer-Pierce." Opp. at 18 (quoting Ning, EX1034, 5:35-38); EX1039 ¶169. Nowhere does Ning teach or suggest that its detection and tracking system can be implemented on-board a drone. A POSITA would not have understood Ning as disclosing or suggesting such an implementation. As a threshold matter, this cropped quotation by Petitioner

is highly misleading because it omits crucial language in the very Ning sentence from which it is drawn.

252. That sentence states: “the system 200 for vision-centric deep-learning-based **road situation analysis** can be a separated system that can be []mounted to any suitable vehicle.” A POSITA would not have found a “road situation analysis” system to be relevant to a drone for many reasons, including the significantly different operating environment and timing constraints on computations. *See* EX2025 (my Initial Declaration), ¶¶71-78. Drone environments are full three dimensional while the ground vehicle environments have a lower dimensionality because the vehicle is driving on the ground plane, where there is no substantial movement in the z-direction.

253. Moreover, Ning describes its systems as “mobile advanced **driver assistance** systems (ADAS).” EX1034, 4:23-28. Ning makes no mention of drones or any airborne vehicles. *See generally* EX1034. Rather, throughout the specification, Ning refers numerous times to road vehicles, pedestrians, and road signs. In light of Ning’s disclosure, a POSITA would have understood that Ning’s system is designed for road vehicles and not for timing-constrained processing on drones operating in a more compute-intensive three-dimensional environment.

d The BLPN combination renders Beard inoperable for its intended purpose

254. Like the BLPKK combination, BLPN combination would also likely render Beard inoperable for its intended purpose. *See McGinley*, 262 F. 3d 1354. Beard requires real-time image processing. *See supra* § VIII.C.1.e; EX1009, 1:56-59, 3:26-30, 3:30-35. It was well understood in 2017, that even if a neural network could somehow be repackaged and executed on the sort of generic image processor taught by Beard, that processor would *still* be unable to execute a neural network in real-time as absolutely required for any computer program providing image-based navigation on a drone. *See* EX2045 at 1-2 (YOLO “cannot work in constrained environments”); EX2051 at 1-2 (YOLO is “too large for many edge and mobile scenarios” such as on-board a drone); EX2052 at 3 (Tiny-yolo and not YOLO could yield reasonable results using limited hardware); EX2044 at 6-7 (the “full YOLOv2 ... is a bit too big to run fast enough on current [i.e., 2017] iPhones”).

255. As such, configuring Beard’s image processor to implement Laufer’s, Pierce’s, or Ning’s neural networks would prevent that processor from performing *real-time* image analysis. This would render Beard’s system inoperable for its intended purpose.

e Ning Does Not Disclose a Neural Network for Image Analysis

256. The claim terms “first neural network” and “second neural network,” as recited in SC 36, are properly understood as neural networks that can be applied to and process images. I am informed by Counsel and understand that “a claim term should be construed consistently with its appearance in other places in the same claim or in other claims of the same patent.” *Rexnord Corp. v. Laitram Corp.*, 274 F.3d 1336, 1342 (Fed. Cir. 2001); *Omega Engineering, Inc v. Raytek Corp.*, 334 F.3d 1314, 1334 (Fed. Cir. 2003). Per claim element [36.d], the “first neural network” has a greater robustness than the “second neural network.” SC 37 also recites a “first neural network” and “second neural network,” and per claim element [37.e2] the “first neural network” of SC 27 also has a greater robustness than the “second neural network” of SC 37. As such, the terms first and second neural network of SC 36 have the same meanings, respectively, as the terms first and second neural network of SC 37.

257. Claim element [37.e3] recites “second neural network was ... applied to the immediately preceding image” and claim elements [37.e3] and [37.e4] both recite “applying the second neural network to the current image.” Thus, the second neural network of SC 37 must be of a type that can be applied to an image. Because the second neural network of SC 36 has the same meaning, that neural network is also of a type that can be applied to an image.

258. Ning’s LSTM (or any LSTM, in general) is not a type of neural network that can be applied to an image.⁸ In fact, Ning states that an “LSTM network is to learn from experience to classify, process and predict time series when there are very long time lags of unknown size between important events.” EX1034, 4:65-67. An image is not a time series; it is a spatial series. Ning itself acknowledges this distinction. *Id.* 12:66-13:2 (“[b]y contemplating a series of frames with LSTMs, where each frame is visually interpreted by convolutional layers [of the YOLO], the network is able to utilize both temporal and spatial information efficiently”). Accordingly, Ning does not teach or suggest applying its LSTM to an image. Rather, Ning applies a YOLO to the images. See EX1034, 4:55-58, FIG. 2.[n1.1]

259. A POSITA would have readily understood that an LSTM cannot be applied to an image without being coupled to a series of convolutional layers for image interpretation. As such, a POSITA would not even have considered an LSTM, by itself, as a second neural network. Petitioners have not argued that the

⁸ In the context of image processing / computer vision, due in part to the fact that many applications require real-time performance, “applying a neural network to an image” or “processing an image using a neural network” is generally understood as directly applying the neural network to an image. See EX2061 at 1 (Figure 1 (“[o]ur system (1) resizes the input image”)); EX2062 at 6 (describing the first layer as “input image”).

BPLN combination discloses a second neural network that includes Ning's LSTM combined with convolutional layers.

260. In summary, for the BLPN combination:

- (i) a POSITA would have reasonably expected the implementation of any of Petitioner's cited neural networks on Beard's image processor to **fail** (if feasible at all) – not succeed;
- (ii) a POSITA would have understood that the addition of Petitioner's cited two neural networks to Beard's system to render that system inoperable for its real-time flight control intended purpose.
- (iii) does not disclose a second neural network, i.e., a neural network that can be applied to an image.

261. As such, a POSITA would not have made the BLPN combination .

2. The BLPN Combination Does Not Teach or Suggest Each and Every Limitation of SC 36

a Claim elements [36.pre], [36.b1], [36.b2], and [36.d]: “[a] reference aerial vehicle, comprising ... a processor configured to: ... us[e] a first neural network ... and ... a second neural network.”

262. Petitioners and Dr. Singhose allege that Beard's image processor and/or Pierce meet the on-board limitations (“processor ... applying”), citing claim elements [21.b], [21.c]. Opp. at 17-21; EX1039, ¶¶169-178. As discussed for claim element [21.b] however, Beard and Pierce, individually or in combination, do not

teach or suggest “applying at the reference aerial vehicle a machine learning model.” *See supra* § VIII.C.2.a (citing *supra* §§ VIII.C.1.a-VIII.C.1.c). Therefore, Beard and Pierce do not teach or suggest a “reference aerial vehicle, comprising ... a processor configured to: ... us[e] a first neural network ... and ... a second neural network,” as SC 36 recites.

b Claim element [36.d]: “a second neural network at ... a second compute cost, ... the first compute cost being greater ... than ... the second compute cost”

(1) Ning does not disclose two neural networks having a cost relationship recited in claim element [36.d]

263. Petitioners allege that in the BLPN combination, Ning’s YOLO is the first neural network and that Ning’s LSTM is the second neural network. Opp. at 19-20; EX1039, ¶¶175-178. Ning, however, couples the LSTM to the YOLO, which is *essential* to the operation of the LSTM, when used for tracking an object in an image. *See infra* ¶¶274-283. Due to this ineluctable dependence, Ning’s YOLO and LSTM do not meet the first and second neural networks limitations of claim element [36.d]. In fact, Ning’s YOLO and LSTM are indivisible parts of a single neural network that Ning describes as the ROLO, and thus do not meet the first and second neural networks limitations of claim element [36.d]. To understand why, I first provide a brief overview the detection and tracking systems disclosed in the ’711 patent.

264. The systems of the '711 patent are designed to minimize the computation cost, i.e., the time required to perform a computation, when the drone is in a target-tracking phase (i.e., “incrementally predicting the location of the target in the image”), because these computations are performed to follow a target that is moving fast in a three-dimensional space. The platform where these computations are performed is also on-board a UAV, and is therefore moving. *See* EX1001, 3:15-24, 3:27-30. In a computer-vision based system, to follow a target, it is necessary to first detect the target in a captured image (or a frame). Once the target is detected, a visual lock should be maintained on the target across subsequently captured images, so that the UAV can be maneuvered to follow the target. *See id.* 21:67-22:2 (“Once an object is *detected* in high-resolution mode (i.e. using DNN or manual input) a tracker algorithm takes control and *maintains lock* on the target.”).

265. The '711 patent describes these two tasks as: (i) detecting, and (ii) tracking or incrementally predicting the target’s three-dimensional location. In particular, the '711 patent states:

Applying different *tracking* algorithms and DNN based *detections* of a target in the video frame provides robustness at a high compute cost. A reduced resource background *tracker may incrementally predict the location of a target in the frame* with low compute cost and average robustness.

Id. 22:2-7.

266. A process tasked with detecting a target in an image does not know *where* (if at all) the target may be located *within* the four corners of the image. As such, it is often necessary to process the entire image. This makes the detection task computationally extensive. Once a target is detected, e.g., in a series of captured images, its movement can be inferred.

267. At the time a target is located, and the tracking phase can commence. The tracking phase has the benefit of the knowledge of the target's location in the image, and as such, tracking need not be performed on the entire image and/or may be performed on a low-resolution image. As such, the per-frame / per-image computation cost of tracking is generally lower than that of detection, i.e., the time required to predict the movement given one image is typically substantially less than that required to detect the target. The target may not always move predictably, however. An unexpected target movement can cause the tracking system to lose track, i.e., not be able to locate the target in an image. The system must then revert back to detecting, and can subsequently switch back to tracking.

268. The '711 patent discloses several embodiments of the above-described process, stating:

Described herein are means for *optimizing* input of video data that is processed to *efficiently locate and track* items of interest using

different processing schemes in view of effectiveness confidence and local computing resources.

Id. 22:11-14.

269. The '711 patent further states:

An example implementation includes method comprising optimizing resources processing a video data stream from a mobile capture device *using a set of processing schemes to track* one or more items of interest. A performance score is associated with each processing scheme confidence to track the one or more items of interest. The method includes *repeatedly determining an active processing scheme* based on the processing scheme with the highest performance score from the set of processing schemes. In response to the performance score of the active processing scheme failing to satisfy a threshold, the method *selects another processing scheme* to process the video data stream. Processing the video data stream can include *identifying [i.e., detecting]* one or more items, classifying each of the items; and *tracking* one or more of items as an item of interest based on the classification.

Id. 22:18-32; *see id.* 23:12-20 (“an *active processing scheme is repeatedly determined* based on the processing scheme with the highest performance score from the set of processing schemes”).

270. A common theme of these disclosures is that a high robustness, high cost first neural network is used to detect a target in an image, and a low robustness, low cost second neural network is used to track / incrementally predict

the target's location in the image. In particular, the '711 patent states: "[P]rocessing schemes can be grouped as: (i) High cost (low refresh rate), high robustness algorithm" "(ii) Medium cost, medium robustness algorithms," "or (iii) Low cost (high refresh rate), low robustness algorithms." *Id.* 27:53-64; *see id.* 28:40-29:3 (describing several neural networks belonging to the three groups).

271. Saying the cost of the second neural network is "low," means that its refresh rate is high, i.e., it updates its understanding of the location of the target within the image faster than the first neural network. *See id.* 27:53-64; EX2033 at 37:6-16. The utilization of the two networks, as described above substantially decreases the average time of detection and tracking together, which allows maintaining a visual lock on, and following a fast-moving target. In particular, the '711 patent states:

Once an object is *detected* in high-resolution mode (i.e. *using DNN* or manual input) a *tracker algorithm takes control* and maintains lock on the target. Applying different tracking algorithms and DNN based detections of a target in the video frame provides robustness at a high compute cost. A reduced resource *background tracker may incrementally predict the location of a target in the frame with low compute cost and average robustness.*

Id. 21:67-22:7; *see id.* 21:60-63 ("[d]eep neural networks (DNN) may be executed to *periodically detect* objects").

272. In order to operate in the manner described above, both the high cost and low cost neural networks must each be capable of *independently* processing an image. See id. 21:67-22:2 (“Once an object is detected ... using DNN .. a tracker algorithm takes control”), 22:23-26 (“[t]he method includes *repeatedly determining an active processing scheme*”). Accordingly, in my opinion, in light of the disclosures of the ’711 Patent’s specification, a POSITA would understand the claimed “a first neural network” of a “first compute cost” and “a second neural network” of “a second compute cost,” as recited in SC 36, as requiring independent neural networks, i.e. where one can be the “active” network without having to depend on the other network.

273. A co-dependence between the two neural networks is inconsistent with the above-described processing scheme that the ’711 patent discloses and claims in proposed substitute claim 36. In particular, if even after the initial object detection, the tracker (second neural network) depends on the detector (first neural network) being active (as is the case with Ning’s LSTMs) the total time (i.e. cost) required to complete a tracking operation for a given image would be the processing time of the first neural network plus the processing time of the second neural network. In this case, no matter how fast the second neural network is in isolation, the tracking time (i.e. cost of the second neural network) would always

be greater than at least the detection time (i.e. cost of the first neural network). This is inconsistent with the cost requirements of SC 36's neural networks.

274. In Ning's system, the YOLO and the LSTM are co-dependent; they are inextricably coupled together in a loop in that the LSTM cannot operate by itself to update the location of a target without the YOLO. On an image-by-image basis, Ning's LSTM cannot function on a given image/frame, and in fact cannot even commence its operation on the given image/frame, until Ning's YOLO has completed its operation on that image/frame and has provided input to the LSTM for that image/frame. The LSTM output for that image / frame, in fact for *each* image/frame, is then recycled by providing that output to the YOLO, forming a loop.

275. In particular, Ning states: in "some embodiments, the Recurrent YOLO (ROLO) can *connect* the convolutional layers of YOLO *to* a recurrent neural network (RNN) long short-time memory (LSTM), for the purpose of real-time object *detection and tracking*." EX1034, 4:55-58, FIG. 3. Ning further states:

The deep neural network ROLO engine 220 includes a deep *convolutional neural network 221* which comprises obstacle detection module 222, pedestrian detection module 224, and traffic sign recognition module 226. The ROLO engine 220 further includes verification and refinement module 227, and multiple *long short-time memory units (LSTMs) 229*.

Id. 5:46-54, FIG. 2. “The visual information” from a camera 211 is “processed by the deep convolutional neural network 221, whose task is generic *object localization and recognition* including pedestrian *detection*, vehicle *detection*, and traffic sign recognition, etc.” *Id.* 5:55-61. The “*tracking* process is performed by multiple long short-term memory (LSTM) units.” *Id.* 8:4-5.

276. More specifically per Ning, “[t]here are two streams of data flowing into the LSTMs, namely, the feature representations from the convolutional layers and the detection information $L_{t,i}$ from the fully connected layers.” *Id.* at 10:34-37, FIG. 8.

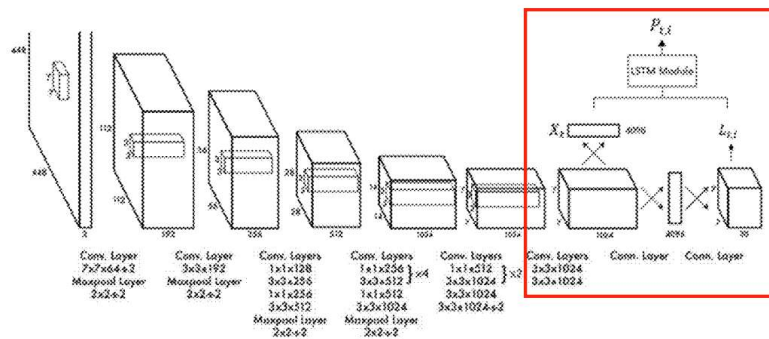


FIG. 8

Id., FIG. 8 (annotated to show the YOLO’s convolutional layers supplying inputs to the LSTM)

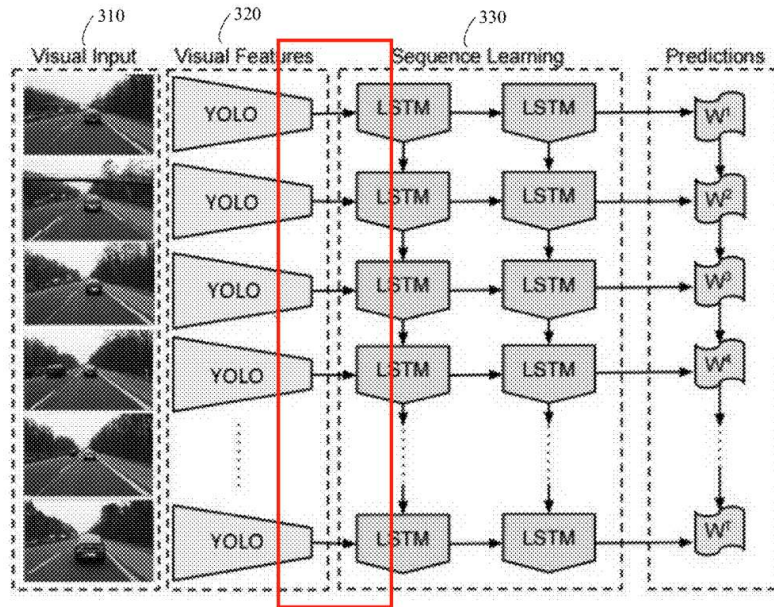


FIG. 3

Id. FIG. 3 (annotated to show that the LSTM receives inputs from the YOLO).

277. Notably, “at each time-step t ,” Ning’s LSTMs receives “a one-dimensional vector of length 4096” which is “extracted” from “vectors output by the last convolutional layer [of the YOLO]” being “forwarded into the first fully-connected layer learned by the pre-training model.” EX1034 at 10:38-54. That is, rather than independently operable neural networks with distinct compute costs, Ning’s LSTM and YOLOs operate in lock-step a sub-modules within ROLO: each LSTM must be continuously fed data from YOLO as part of the overall ROLO network.

278. The above-described connection is *essential* to Ning’s ROLO. Ning explains this by stating that an “LSTM network is to learn from experience to classify, process and predict *time series* when there are very long time lags of

unknown size between important events.” *Id.* 4:65-67. An image is not a time series; it is a spatial arrangement of pixel data. Ning itself acknowledges this distinction. *Id.* 12:66-13:2 (“[b]y contemplating a series of frames with LSTMs, where *each frame is visually interpreted by convolutional layers* [of the YOLO], the network is able to utilize both temporal and spatial information efficiently”).

279. Ning also describes that:

The disclosed system 200 uses a tracking module [i.e., the LSTM 229] in the deep neural network ROLO engine 220, together with a Kalman filter to follow detected pedestrians/vehicles over time. This step has several purposes: avoiding false detections over time, predicting future positions, thus *feeding the object recognition modules* including obstacle detection module 222, pedestrian detection module 224, and traffic sign recognition module 226, *with pre-candidates*, and, at a higher level, making useful inferences about pedestrian or vehicle behavior (e.g., walking direction, or lane change intention).

EX1034 at 6:33-43. Thus, in addition to receiving inputs from the YOLO, the LSTM also provides “pre-candidates” to the various detection modules of the YOLO, arranging the YOLO and the LSTM in a co-dependent loop.

280. In this configuration, the cost of the LSTM (alleged second neural network) in operation is not less but greater than that of YOLO (alleged first neural network). The LSTM’s individual cost alone cannot be understood as its cost

because in Ning's ROLO, it is *infeasible* to operate the LSTM, by itself.⁹ An LSTM cannot process an image by itself. At least one, and typically several convolutional layers, that Ning's YOLO provides, are *essential* to the LSTM's operation. *See id.* 12:66-13:2. Thus, given the ROLO's essential loop structure, the cost of Ning's LSTM (alleged second neural network) is greater than the cost of Ning's YOLO (alleged first neural network). This is the exact opposite of what claim element [36.d] recites. Accordingly, Ning teaches or suggests neither a first neural network nor a second neural network having a cost relationship recited in claim element [36.d].

281. A POSITA would have recognized that from Ning's ROLO, one cannot simply extract the LSTM and use it independently for object tracking in images. This is because LSTM were generally understood as unsuitable for this task. For example, Shi reported that "[a]lthough the FC [fully connected]-LSTM layer has proven powerful for handling temporal correlation, it contains *too much redundancy* for spatial [i.e., image] data." EX2105 at 3. In reporting the performance of the FC-LSTM on predicting cloud movement, Shi states:

⁹ To the extent Ning's LSTMs individual cost, ignoring the ROLO configuration, may be considered, that, too is greater than that of the YOLO, as discussed below. *See infra* § VIII.E.2.b(2).

the performance of the FC-LSTM network is *not so good for this task*, which is mainly caused by the strong spatial correlation in the radar maps, i.e., the motion of clouds is highly consistent in a local region. The fully-connected [LSTM] structure has too many redundant connections and *makes the optimization very unlikely* to capture these local consistencies.

Id. at 8. To address this problem, Shi presented a solution similar to Ning’s – coupling the LSTM to a convolution layer that first analyzes the spatial data in images. *Id.* at 3 (“we propose an extension of FC-LSTM which has convolutional structures in both the input-to-state and state-to-state transitions”).

282. Shi’s observations were confirmed later, as an *inherent* drawback of LSTMs:

However, there is an *inherent drawback in LSTM*, in which the *spatial structure information [in images] will be lost* when unfolding the input to the 1-D from. To better model the spatiotemporal relationships, Shi et al. [] extended this data processing method in LSTM to the convolution operation and proposed Convolutional LSTM (ConvLSTM), which consists of *convolution structures in both the input-to-state and state-to-state transitions*.

EX2104 at 3, col. 2.

283. Accordingly, a POSITA would have understood that Ning’s LSTM cannot be separated from its coupling to the YOLO.¹⁰ Moreover, Ning’s YOLO also depends on the pre-candidates provided by the LSTM forming a loop. The loop structure, and in particular this inability to decouple LSTM from YOLO for any frame/image, effectively reduce Ning’s structure into a single combined neural network (i.e., ROLO). Thus, a POSITA would have understood that while Ning uses a YOLO and an LSTM, for object detection and tracking, Ning discloses only a single, unified neural network, ROLO, and not two neural networks as claim element [36.d] recites.

284. To the extent Ning discloses two neural networks, a POSITA would have understood, as described below, that Ning’s YOLO and LSTM are not the first and second neural networks having the required cost relationship, i.e., “a first neural network at a first robustness and a first compute cost... a second neural network at a second robustness and a second compute cost” . Therefore, a POSITA would not have been motivated to combine Ning with other references, because

¹⁰ If so separated, for the LSTM to function, it would have to be coupled to another convolutional neural network, as Ning itself recognizes. EX1034, 12:66-13:2 (“contemplating a series of frames with LSTMs, where each frame is visually interpreted by convolutional layers”).

such a combination could not be reasonably expected to achieve the claimed invention.

285. In summary, Ning does not teach or suggest the first and second neural networks that claim element [36.d] recites, and a POSITA would have understood that Ning's YOLO and LSTM cannot be configured to operate as claim element [36.d] recites.

(2) BLPN does not disclose or render obvious a first neural network for “detecting of the target in the image” having a greater cost and robustness than a second neural network for “determining of the three-dimensional relative location of the target in the image”

286. Even if Ning disclosed the claimed two neural networks (it does not, as discussed above), it does not disclose the particular configuration of a first neural network and a second neural network as recited in claim 36. As noted, Petitioners allege that in the BLPN combination, the YOLO is the first neural network and that the LSTM is the second neural network. Thus, it follows that to meet the limitations of claim element [36.d], the compute cost of Ning's YOLO must be greater than that of the LSTM. The '711 patent also describes that a high cost processing scheme has a low refresh rate and that a low cost processing scheme has a high refresh rate. EX2019 ¶¶145-150; EX1001, 27:53-28:47. Dr. Singhose agrees. EX2033 at 37:6-16. It then follows, that if YOLO is the high cost, “first” neural network, and LSTM is the low cost “second” neural network,

the refresh rate of YOLO must be less than that of LSTM. The opposite however, is true.

287. Ning is not an outlier in reversing the robustness and compute cost limitations taught by the 711 patent. Lu reported a system similar to Ning's. Ning used a sequence of YOLO and LSTM. EX1034, 10:33-37 (describing "two streams of data flowing into the LSTM" sequence from the convolution and fully connected layers of the YOLO), FIG. 3. Likewise, Lu used a sequence of a single shot detector (SSD) and an LSTM sequence. EX2054 at 7 (p. 2350) (disclosing "front-end one shot SSD"), 3 (p. 2346), Figure 2 (depicting, in each row, two LSTMs in sequence following an SSD). A side-by-side comparison of the two systems is shown below.

EX1034 – Ning Patent

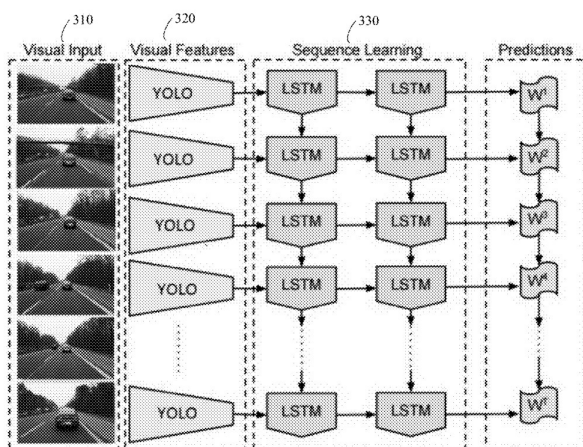


FIG. 3

EX1034, FIG. 3

EX2054 – Lu Article

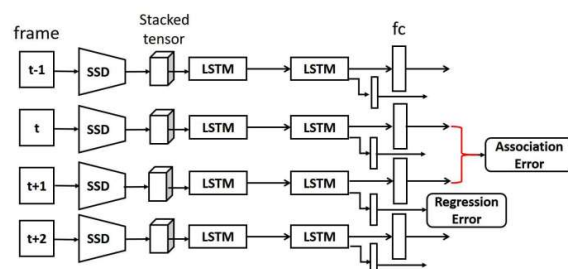


Figure 2. Our LSTM architecture in detail. We solve the two complementary tasks, namely regression and association in a unified framework by defining the regression error and association error.

EX2054 at 3 (p. 2346), Figure 2

288. Lu states: “[o]ur implementation is coded on an Intel I7 4.0 GHz PC with one **Titan X 12 GB GPU** and 32GB memory. The front-end one shot SSD runs 33 FPS, while the proposed association LSTM runs 12 FPS on both datasets in the testing stage, which is nearly real time.” EX2054 at 7 (p. 2350). Redmon reported that YOLO, also using the Titan X GPU, runs at 45 frames per second. EX1036 at 48 (p. 779) (“YOLO is extremely fast” and that “[o]ur base network runs at **45 frames per second** ... on a **Titan X GPU**”). Because Redmon and Lu used the same Titan X GPU, the speed-performance they reported are comparable.

289. Thus, as Lu and Redmon reported, the YOLO, at 45 fps, is the fastest and the LSTM at only 12 fps on the same processor, was much slower. This is expected; LSTM was generally known as effective for time-series analysis, but a relatively slow processing scheme. But, as noted above, this result is exactly the opposite of what claim element [36.d] requires. The “first” neural network has a higher compute cost, i.e., a lower refresh rate than that of the “second” neural network, which has a lower compute cost, i.e., a higher refresh rate. In the system of the BLPN combination, the YOLO (the first neural network) has a higher, not lower, refresh rate than that of the LSTM (the second neural network).

290. As such, to the extent the YOLO is the first neural network, the LSTM is not the second neural network at ... a second compute cost, ... the first compute cost being greater ... than ... the second compute cost,” as SC 36 recites.

c Claim element [36.d]: “incrementally predicting the location of the target in the image using a second neural network”

291. Ning does not teach or suggest this limitation. In light of the specification of the '711 patent, this limitation is properly interpreted as processing the image, i.e., the image captured per claim element [36.a], using “a second neural network.” This is because only that interpretation is consistent with the reasons the '711 patent provides for using not one but two neural networks, and also because in all of the embodiments that the '711 patent discloses, the second neural network processes captured images.

292. The '711 patent states:

On a fixed compute budget there is a **negative relation** between robustness of an algorithm and the cost and/or latency of the algorithm per spacio-temporal frame. Tight restrictions on available compute cycles limit the responsiveness of the system, especially on autonomous or semiautonomous drones. By combining multiple algorithms in a way that achieves **high combined frame rate** with higher quality prediction, the system improves outcomes on a limited resource budget.

EX1001, 21:52-60.

293. The '711 patent further states: an “example implementation includes method comprising optimizing resources **processing a video data stream** from a mobile capture device using a **set of processing schemes to track** one or more

items of interest.” *Id.*, 22:18-21. Additionally, the ’711 patent states that “[o]nce an object is detected in high-resolution mode (i.e. using DNN or manual input) a tracker algorithm takes control and maintains lock on the target.” *Id.* 21:67-22:2. To this end, a “reduced resource background tracker may incrementally predict the location of a target **in the frame** with low compute cost and average robustness.” *Id.* 22:5-7; Paper 21 (MTA) at 18; EX2025 ¶170; EX2019 ¶110.

294. Thus, the ’711 patent explains that while high-resolution processing can be effective in detecting a target in an image accurately, such processing has a high cost in terms of latency. This would decrease the rate at which frames are processed. As such, rather than continually using the high-resolution processing, upon the initial detection, a lower cost but fast, high refresh rate tracker is used. That tracker “incrementally predict[s] the location of a target **in the frame**,” i.e., the tracker processes the captured frames or images.

295. The examples of such trackers include low resolution, low cost, low robustness, but high refresh rate neural networks. *Id.* 27:53-64 (disclosing high cost / low refresh rate and low cost / high refresh rate algorithms), 28:48-29:3 (disclosing “SqueezeNet super fast rejection CNN” as an example of low cost / high refresh rate processing). Such a low cost neural network, i.e., the claimed second neural network, operating as a tracker, “incrementally predict[s] the location of a target **in the frame**,” i.e., processes the captured frames or images.

296. The '711 patent describes:

The tracking system 1500 comprises a cameras 1528 producing multi-camera views 1518 that are input to a processor 1508. The processor operates to filter and de-noise the multi-camera views 1518 to populate a pixel domain 1516. The **pixel domain 1516 is divided into nodes** (e.g., node 1506, node 1520, node 1514, and node 1512 that are then analyzed by a high resolution tracker 1502. Output of the high resolution tracker 1502 is input to a fast, low power consumption low resolution tracker 1510. In this example, the node 1506 comprises an anomaly 1522. The high resolution tracker 1502 identifies the anomaly 1522 as a detected object 1524, which is then tracked by the low resolution tracker 1510.

EX1001, 24:13-27, FIG. 15; EX2019 ¶121.

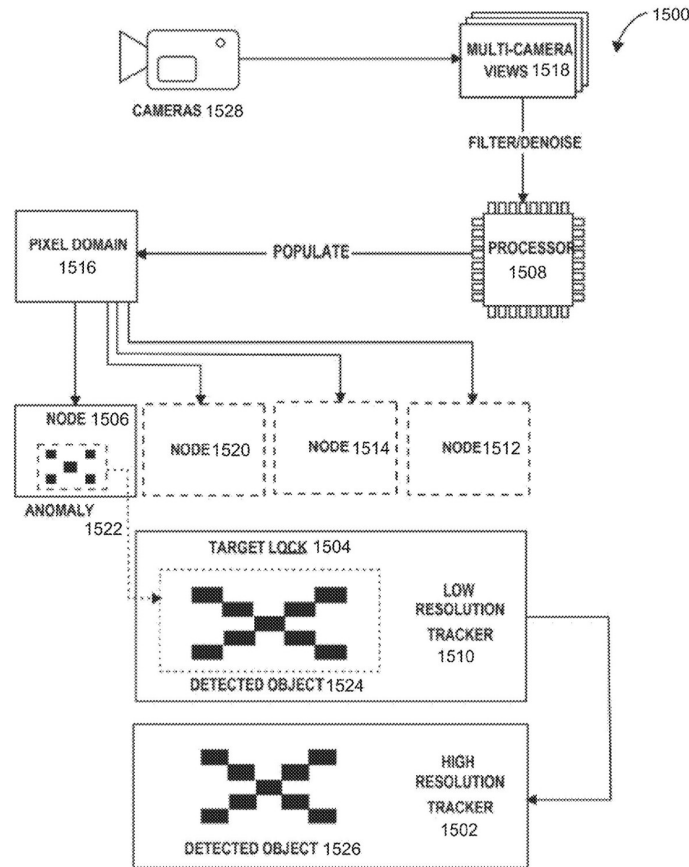


FIG. 15

EX1001, FIG. 15 (depicting high and low resolution trackers processing pixel nodes 1506 that includes the anomaly 1522 detected by the high resolution tracker); EX2019, FIG. 15.

297. Thus, the ‘711 patent’s low-cost tracker, i.e., a “reduced resource background tracker,” tracks the anomaly 1522 previously detected by its high-resolution tracker, by “incrementally predict[ing] the location of a target in the frame with low compute cost and average robustness.” EX1001, 22:3-7. To this end, the low-cost tracker processes a portion of the image in a captured frame. An example high resolution tracker is a DNN, i.e., the claimed “first neural network,”

and an example low compute cost, low resolution tracker is SqueezeNet super fast rejection CNN, i.e., the claimed “second neural network.” EX1001, 27:53-64, 28:40-51, 28:64-29:3; EX2019 ¶¶145, 150-151; MTA at 18-19; EX2025 § XI(D), ¶170. In this example, the low resolution tracker, i.e., the claimed “second neural network,” processes a pixel domain node (which includes an anomaly (e.g., a target) detected by a high resolution tracker), i.e., a portion of the frame captured by a camera.

298. In all other embodiments using both high cost and low cost processing schemes / neural networks, the ’711 patent discloses using low-resolution images. *See* EX1001, 21:23-29 (“the “spacio-temporal awareness engine 1100” includes a “low-resolution converter 1108”), FIG. 11, 24:5-7 (disclosing a “parallel image processing process 1400 [that] comprises the high resolution process 1418 and the low resolution process 1416”), FIG. 14, 24:46-48 (disclosing “a system 1700 for converting **camera input** into a vector **for low resolution tracker 1510**”), FIG. 17. Thus, the claim limitation “incrementally predicting the location of the target in the image using a second neural network” requires applying the second neural network to the captured image, which may include a target previously detected by the high resolution tracker, i.e., the first neural network.

299. In stark contrast, Ning’s LSTM, i.e., the alleged second neural network, is not applied to any image at all. Rather, Ning states: there “are two

streams of data flowing into the LSTMs, namely, the feature representations from the convolutional layers [of YOLO] and the detection information $L_{t,i}$ from the fully connected layers” of YOLO. EX1034, 10:34-37. The detection information $L_{t,i}$ includes the “detections (x, y, w, h, c) at time t, i.e., the location parameters of a bounding box. *Id.*, 10:17-32.

300. The first stream that the LSTM receives, however, is not a stream of the captured images. Instead, these are the features that are extracted by the convolutional layer of YOLO. A POSITA would have known that in the context of neural networks and machine learning in general, “features” are a set of **attributes** associated with the instances of data used for learning or evaluation. EX2060 at 3. These features are not the physical features of a detected target (e.g., rotors, cockpit, landing gear, of a drone). They are also not portions of images themselves, when the instances of data are images. Therefore, while Ning’s LSTM may predict the location of a detected target, it is not “a second neural network” that “incrementally predict[s] the location of the target **in the image**,” as PSC 36 recites.

301. Beard, Laufer, and Pierce do not disclose “a second neural network at a second robustness and a second compute cost” that are respectively lower than the first robustness and the first cost of a first neural network. Petitioners do not allege so. Therefore, the BLPN combination does not teach or suggest

“incrementally predicting the location of the target in the image using a second neural network” as SC 36 recites.

302. In conclusion in connection with SC 36, a POSITA would not have combined BLPN. Regardless, the combination does not teach or suggest all of the limitations of SC 36, and the total disclosure of the combination is not enabling. Therefore, in my opinion, SC 36 is patentable over the BLPN combination.

F. Ground 28 –Pierce, Kumar, Khan, and Beard (the PKKB Combination”) Does Not Render SCs 21, 23-26, and 32 Obvious

303. In the PKKB combination, Petitioners and Dr. Singhose allege that Pierce’s object tracking module 615 is on-board a drone, and that module includes one or more neural networks. This, according to Petitioners shows the limitation of claim element [21.b], i.e., “applying at the reference aerial vehicle a machine learning model to the image.” *See* Opp. at 21-22; EX1039, ¶¶179-184.

304. I have explained above that this allegation is false. A POSITA would have understood that Pierce’s object tracking module 615 is a separate, ground-based component and is not on-board a drone. *See supra* § VIII.C.1.a.

1. A POSITA Would Not Have Combined or Modified PKKB

a A POSITA would have understood that Pierce’s on-board drone system could not simply be programmed to implement Pierce’s neural networks

305. To the extent Petitioners allege that a POSITA would have modified Pierce’s on-board processor so as to implement the neural network(s) from the

ground based object tracking module 615, *see* Opp. at 21-22; EX1039, ¶¶181-184, a POSITA would have known that such an implementation and/or that of any neural networks on Pierce’s on-board processor would have been a significant challenge, if feasible at all, and would not have expected such a modification to succeed. *See supra* § VIII.C.1.b.

b The PKKB combination has duplicative features

306. The PKKB combination results in duplicative features that increase system complexity. In particular, in the proposed combination, even though the neural networks of Pierce and (Kumar, if used) learn features, Khan’s additional HoG-based feature extraction would still also be required, resulting in a redundancy that Petitioners and Dr. Singhose do not explain or justify. *See supra* § VIII.C.1.d.

c The PKBB modification renders Pierce inoperable for its intended purpose

307. Pierce’s system is designed for real-time processing and target tracking. For example, Pierce states: “including the targets linear velocity in determining the linear velocity 411 of the drone allows the system to predict the moving targets movement and velocity.” EX1005, ¶36. Pierce further states:

[S]uch predictive capability may allow the drone to change direction and/or speed to correspond to changes in direction and/or speed of the moving target in **real-time**. Thus, in some embodiments, the drone is

able to slow down [] in **real-time** and not overshoot the moving target if the moving target Suddenly stops moving.

Id. ¶37.

308. Pierce does not disclose that its processor is of any particular type, e.g., a GPU, multi-core processors, etc., that is well-suited for executing neural networks. Therefore, as discussed for Ground 23 in connection with Beard, the proposed modification / combination would render Pierce’s system inoperable for its intended use. *See McGinley*, 262 F. 3d 1354; *supra* § VIII.C.1.e.

309. For the foregoing reasons, a POSITA would not have modified Pierce (to the extent Pierce is modified as discussed above) or have made the proposed combination.

2. The PKKB Combination Does Not Teach or Suggest Each and Every Limitation of SC 21

a Claim element [21.b]: “applying at the reference aerial vehicle a machine learning model”

310. A POSITA would have understood that Pierce’s object tracking module 615 is a separate, ground-based component, and not on-board a drone. Moreover, contrary to Petitioners’ allegation, Pierce’s on-board processor (if any) could not be modified to implement the object tracking module’s neural network(s). *See supra* §§ VIII.C.1.a, VIII.C.1.b. Therefore, Pierce does not teach or suggest this claim element.

b [21.b] “applying ... a machine learning model to the image ... to identify a bounding box outlining a portion of the image that includes detected features of the target”

311. As they did for the BLPKK combination, Petitioners cite Khan and Kumar as allegedly disclosing this limitation. Opp. at 21-22; EX1039, ¶¶182-184. Therefore, for the reasons discussed in connection with the BLPKK combination, the PKKB combination also does not teach or suggest this claim limitation. *See supra* § VIII.C.2.b.

312. In summary, for the foregoing reasons, a POSITA would not have made the PKBB combination and/or modified Pierce. Regardless, the PKKB combination does meet all of the limitations of SC 21, and hence, does not render SC 21, and its dependent claims SCs 23-26, and 32, obvious.

G. Grounds 29 and 30 – Further Combinations with Laufer and Hogasten Do Not Render SCs 22, 29, 34, and 35 Obvious

313. Grounds 29 and 30 challenge validity of dependent SCs 22, 29, and 35 (Ground 29), and 34 (Ground 30), by combining Laufer and Hogasten, respectively, to the PKKB combination. Opp. at 24; EX1039, ¶¶194-196. These claims depend from SC 21 and are patentable at least because SC 21 is patentable, as discussed above. *See supra* § VIII.E.

H. Ground 31 –Pierce, Kumar, Ning, and Beard (the “PKNB Combination” Does Not Render SC 36 Obvious

1. A POSITA Would Not Have Combined PKNB

314. In the Pierce, Kumar, Ning, and Beard, combination, Petitioners and Dr. Singhose allege that Pierce’s on-board processor can be modified to implement Ning’s YOLO. Opp. at 24-25; EX1039, ¶¶197-203. They do not cite any evidence showing that such a modification would have been within the scope of a POSITA. In fact, a POSITA would have known this to be extremely challenging, if feasible at all. This is because, like Beard, Pierce does not disclose that its processor 601 is any particular kind of processor, such as a GPU, a multi-core processor, etc. *See* EX1005 ¶42; *see generally id.* Thus, a POSITA would understand Pierce’s processor as a generic processor / CPU.

315. For the BLPN combination, I have explained the substantial difficulty a POSITA would have faced in undertaking the **three-step reconfiguration** necessary to implement Ning’s YOLO on Beard’s generic image processor. In implementing Ning’s YOLO on Pierce’s processor, a POSITA would have expected the same challenges. Petitioners do not explain how a POSITA would have accomplished this substantial modification. For those reasons, as discussed for the BLPN combination, a POSITA would not have made the PKKNB combination. *See supra* § VIII.E.1.

2. The PKNB Combination Does Not Teach or Suggest Each and Every Limitation of SC 36

316. As discussed for the BLPN combination, neither Beard nor Pierce teaches or suggests “[a] reference aerial vehicle, comprising ... a processor configured to: ... us[e] a first neural network ... and ... a second neural network.” Moreover, Ning does not teach or suggest “incrementally predicting the location of the target in the image using a second neural network.” Therefore, for the reasons provided for the BLPN combination, the PKNB combination also does not teach or suggest each and every limitation of SC 36. *See supra* § VIII.E.2.

317. For the forgoing reasons, the PKNB combination does not render SC 36 obvious.

I. Grounds 32-34 – Further Combinations with Sahasrabudhe, Laufer, and Hogasten Do Not Render SCs 21-26, 29, and 31-34 Obvious

318. Ground 32 adds Sahasrabudhe to the PKKB combination, and challenges validity of SC 21 and dependent SCs 23-26 and 32. Ground 33 further adds Laufer to the PKKB and Sahasrabudhe combination, and challenges dependent SCs 22, 29, and 35. Ground 34 further adds Hogasten to the PKKB and Sahasrabudhe combination, and challenges dependent SC 34. Opp. at 25; EX1039, ¶¶204-206. These claims depend from SC 21 and are patentable at least because SC 21 is patentable, as discussed above for the PKKB combination. *See supra* § VIII.E.

J. Ground 35 – Pierce, Kumar, Sahasrabudhe, and Ning (the “PKSN Combination”) Does Not Render SC 36 Obvious

319. Petitioners allege that the PKSN combination renders SC 36 obvious for the reasons they provided for Ground 32 – the PKNB combination. Opp. at 25; EX1039, ¶¶207-208. For the reasons provided herein for the PKNB combination, the PKSN combination does not render SC 36 obvious. *See supra* § VIII.H.

IX. CONCLUSION

320. For the foregoing reasons, it is my opinion that the challenged claim 12, and proposed substitute claims 21-37 are patentable.

321. I currently hold the opinions that I have expressed in this declaration. My analysis may continue, however, as I may acquire or receive additional information or insights that may result in revised or supplemental observations or opinions.

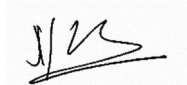
322. I understand this declaration may be used as evidence in the instant IPR proceeding the PTAB.

323. I declare that all statements made herein on 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 or Title 18 of the United State Code, and that such willful false statements may jeopardize the validity of the '711 patent.

324. I declare under penalty of perjury that the foregoing is true and correct.

Dated: June 22, 2026

Respectfully submitted,

A handwritten signature in black ink, appearing to be 'N. Papanikolopoulos', written on a light gray rectangular background.

Dr. Nikolaos Papanikolopoulos