

Patent Owner's Demonstratives for Oral Hearing

Case No. IPR2025-00629

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

July 28, 2026

List of Exhibits Cited

EX1001	EX2001	EX2043
EX1003	EX2002	EX2044
EX1005	EX2025	EX2047
EX1010	EX2033	EX2049
EX1021	EX2036	EX2054
EX1034	EX2039	
EX1036	EX2042	

DEMONSTRATIVE EXHIBIT – NOT EVIDENCE

Invention of the '711 Patent

- Was funded by the Department of Defense (War)
- Accepted as an Exhibit at the Smithsonian

'711 Patent

(57)

ABSTRACT

A flight control operation of a reference aerial vehicle is performed. For example, an image captured by an image sensor of the reference aerial vehicle is received. A target is detected in the image. A three-dimensional relative location of the target with respect to the reference aerial vehicle is determined based on the image. The flight control operation is performed based on the three-dimensional relative location of the target with respect to the reference aerial vehicle.

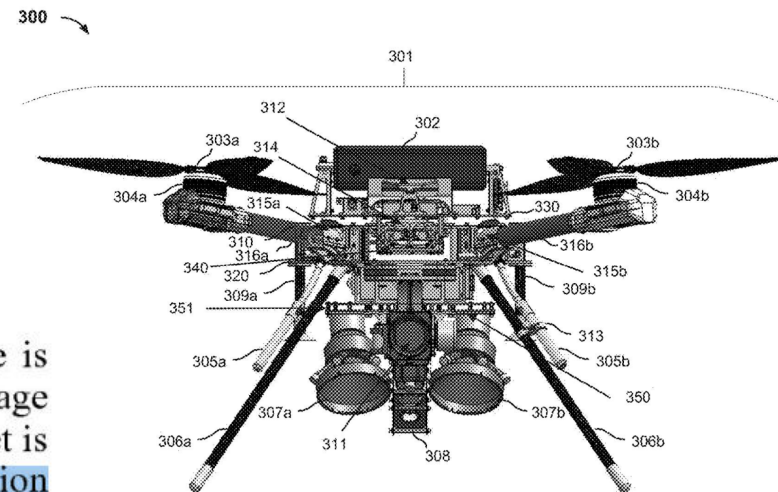


FIG. 3A

Paper 20 at 6-8; EX1001 Abstract, FIG. 3A; EX2001; EX2002

Key Problems the '711 Patent Solves

- No human control
- No reliance, in substance, on ground communications

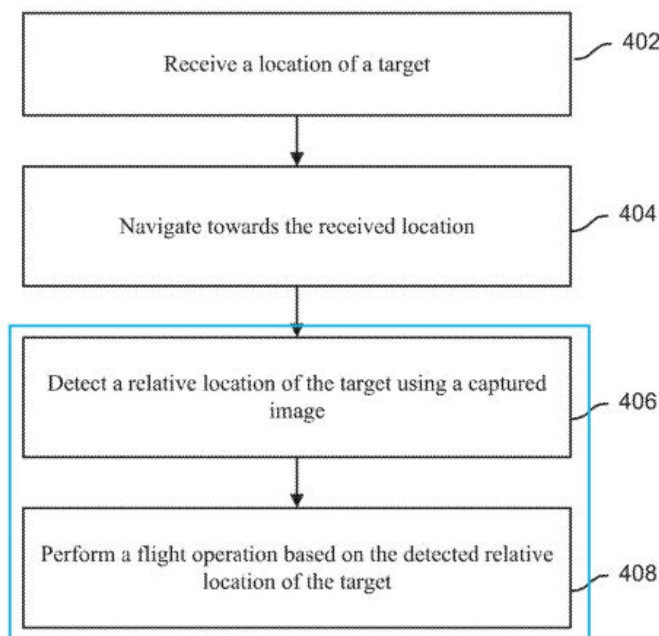
'711 Patent

Typically unmanned aerial vehicles and drones are controlled using flight controllers manually operated by a human user. This requires the human user to have a visual sight of the aerial vehicle and/or an image captured by the aerial vehicle. However, this may be challenging in circumstances where a trained human user is unavailable or unable to react quickly enough to perform a desired flight maneuver. Additionally if communication between the aerial vehicle and the flight controller is lost or becomes unreliable, the aerial vehicle may be unable to complete a desired flight operation.

Paper 20 at 6-8, 14-15; EX1001 1:27-37

Solution 1: On-Board Machine Vision

'711 Patent



In some embodiments, an image is received and a target aerial vehicle is detected in the image. For example, a stream of images captured by one or more cameras of an interceptor aerial vehicle is analyzed using deep learning, neural networks, and/or other machine learning techniques to recognize a target aerial vehicle in the captured image. A three-dimensional relative location of the target aerial vehicle is determined with respect to a reference aerial vehicle based on the image. For example, azimuth and attitude directions as well as distance from the reference aerial vehicle are determined based on the image. A flight control operation is performed based on three-dimensional relative location. For example, the reference aerial vehicle is guided towards the three-dimensional relative location at a dynamically determined speed.

Paper 20 at 8-9; Paper 28 at 8; Paper 42 at 1; EX1001 3:31-45, FIG. 4

Solution 1A – Improved 3D Location / Distance (SC 21)

- Conventional signal processing or ML techniques are not accurate enough for noisy, 3D environment

'711 Patent

aerial vehicle). The relative distance may be determined based on a size of the bounding box within the entire captured image and a detected classification of the target. For example, the target that is further away will appear smaller in the image as compared to the target that is closer to the reference location, and thus 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 output of the image analysis (e.g., output from applying the machine learning model) may include a bounding box outlining an area within the image that includes the image portion with the target (e.g., bounding box outlines the minimum sized box that includes detected features of the target) and/or a classification of the detected target (e.g., type, model, class, manufacturer, size, etc. of the detected target).

Thus, a measurement of the target within the image (e.g., width of the bounding box, length of the bounding box, diagonal length of the bounding box, area of the bounding box, number of pixels in the bounding box, number of pixels across a dimension of the bounding box, etc.) can be mapped to a distance value based on a determined size corresponding to the detected type/classification of the target. A table

- Tightly coupled, multi-function ML model (SC 21)** provides:

- A bounding box
- Around detected features of the target
- A classification used to determine physical size of the target

- Enabled near stage autonomous operation [Paper 20 at 8-9; EX2025 ¶¶ 71-80; EX1001 17:39-51, 17:31-38, 17:61-67](#)

Solution 1B: On-Board Machine Vision on Limited Compute Budget (SCs 36, 37)

- **Selective Application of two NNs**

- To improve quality and speed
- While under compute budget

- Often, one NN did not work

- Too slow or too inaccurate

'711 Patent

limited resource budget. Deep neural networks (DNN) may be executed to periodically detect objects and distinguish targets. A main high resolution tracking engine executes

domain inconsistencies. 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.

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

In an example implementation, 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 tion method. If the confidence is below low threshold, falls outside of a range, or the validation method fails, group (i) processing is performed. If the confidence is determined to be high, then group (iii) algorithms can be applied to optimize resources. Different Groups of algorithms (e.g.,

Paper 48 at 1; EX2036 ¶¶ 39-40, 82;

EX1001 21:60-62, 21:67-22:7, 21:52-60, 27:65-28:18

Solution 2: Autonomous Navigation

'711 Patent

At 402, a location of a target is received. An example of the location is geographical location coordinate and an example of the target is a vehicle detected to be not allowed in a monitored airspace (e.g., target aerial vehicle 110 of FIG. 1). In some embodiments, the location is a geographical location coordinate that has been determined using one or more remote sensors. For example, the location of target

▪ Autonomous navigation (Claim 12)

- Avoids
 - Need for human control
 - Reliance on continuous ground communication

At 404, the received location is utilized to navigate towards the received location. For example, the location is received at a navigation component and/or a flight controller of an interceptor aerial vehicle and the direction and/or speed of the aerial vehicle is automatically and autonomously adjusted to turn and fly towards the received location in a navigation mode based on the received location. This allows the interceptor aerial vehicle to be approached and become closer to the vicinity of the target. However, because the received location may represent a rough and/or approximate location of the target, the received location may not represent the exact location of a moving target required to be known to successfully deploy an interdiction/capture mechanism.

At 406, a relative location of the target is detected using a captured image. For example, once an image sensor/

Paper 37 at 12, 17; Paper 42 at 1; EX2036 ¶¶¶ ; EX1001 15:32-38, 15:64-16:13



Claim 12 – Autonomous Navigation

Claim 12 – PO’s Construction of “Autonomously Navigating” Is Correct

Petitioners’ Construction	PO’s Construction
“navigated and maneuvered by a computer, without a need for human control or intervention under normal road conditions”	“making navigation decisions by the reference aerial vehicle by-itself, without outside control”

- Per Petitioners’ construction:
 - The computer can be remote from the drone, e.g., at a ground station
- But specification distinguishes autonomous from such remotely controlled operations

'711 Patent

a manner that is consistent with the flight commands. In some embodiments, the CPU is configured to receive flight instructions from a remote command center. In other embodiments, the CPU is configured to autonomously fly UAV 301.

ver. Additionally if communication between the aerial vehicle and the flight controller is lost or becomes unreliable, the aerial vehicle may be unable to complete a desired flight operation.

Paper 20 at 7, 10-12; Paper 28 at 11-12; Paper 42 at 2, 12; EX1001 9:47-51, 1:34-36

Claim 12 – PO’s Construction of “Autonomously Navigating” Is Correct

- PO’s construction is derived from the specification

'711 Patent

of UAV 301. The CPU is configured to control the overall operation of UAV 301. The CPU may be coupled to a

311. The CPU may be configured to perform one or more flight decisions for the UAV. For example, the CPU may provide one or more flight commands to a flight controller module. For example, a flight command may include a specified speed for the UAV, a specified flight height for the UAV, a particular flight path for the UAV, etc. In response to the one or more flight commands, the flight controller module is configured to control the motors associated with the UAV (e.g., motors 304a, 304b) so that UAV 301 flies in a manner that is consistent with the flight commands. In

embodiments, the CPU is configured to autonomously fly UAV 301.

Computing chassis 302 is configured to protect the CPU of UAV 301. The CPU is configured to control the overall operation of UAV 301. The CPU may be coupled to a plurality of computing modules. For example, the plurality of computing modules may include an interdiction control module, an image processing module, a safety module, a flight recorder module, etc. The CPU may provide one or more control signals to each of the plurality of computing modules. For example, the CPU may provide a control signal to the interdiction control module to activate one of the net launchers 307a, 307b to deploy a net. The CPU may provide a control signal to the image processing module to process an image captured by the visual detection system 311. The CPU may be configured to perform one or more flight decisions for the UAV. For example, the CPU may provide one or more flight commands to a flight controller module. For example, a flight command may include a specified speed for the UAV, a specified flight height for the UAV, a particular flight path for the UAV, etc. In response to the one or more flight commands, the flight controller module is configured to control the motors associated with the UAV (e.g., motors 304a, 304b) so that UAV 301 flies in a manner that is consistent with the flight commands. In some embodiments, the CPU is configured to receive flight instructions from a remote command center. In other embodiments, the CPU is configured to autonomously fly UAV 301.

Paper 20 at 10-11; Paper 42 at 1-2, 9, 23; EX1001 9:26-27, 9:38-47, 9:50-51, 9:25-51



Substitute Claims

Written Description Support

Specification Fully Supports SC 37

SC 37

“the detecting of the target in the current image further comprises **making a selection**, from between the first and the second neural networks ... **based on a confidence level for a detection** of the target in the current image”

- Petitioners argue that the '711 patent does not teach “decision making based on ... the current image”
- SC 37 does not recite and does not require decision making based on the current image
- Selection is based on a confidence level “**for** a detection”
- **Not** on a confidence level **based on or derived from** a detection of the target in the **current image**
- SC 37 does not recite such a limitation and does not require such an operation
 - **Decision is not based on the yet to be processed current image**

Paper 46 at 1-2; Paper 48 at 1-2

Specification Fully Supports SC 37: Image-by-Image Decision Making

- Specification teaches
 - Image-by-image decision making
 - Decision based on confidence level from applying NN to the immediately preceding image
- This is the selective application of two NNs discussed earlier
 - On a per spacio-temporal frame basis – meeting the two requirements above

'711 Patent

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

The spatial-temporal (spacio-temporal) awareness engine utilizes the limited resources available on an autonomous robotic systems (ARS) (e.g. drones, self driving cars, etc.). The spacio-temporal awareness engine utilizes a multi-camera view which is processed in parallel by a cascade of noise removal and super pixel feature consolidation and isolation algorithms to bring them to lower resolution images. These images are processed continuously using

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. The confidence num-

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

Paper 37 at 21-25; Paper 48 at 1; EX2036 ¶¶ 39, 65-87, 292; EX1001 21:52-60, 21:35-42, 27:29-32, 22:21-29

Specification Fully Supports SC 37: Thresholds and Range

SC 37

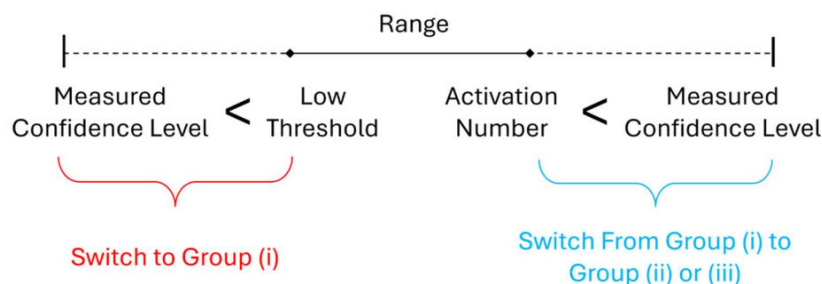
“a first threshold of a range defined by the first threshold and a second threshold greater than or equal to the first threshold”

- Specification teaches a “low threshold,” a “range,” and an “activation number” (as a high threshold)

'711 Patent

tion method. If the confidence is below low threshold, falls outside of a range, or the validation method fails, group (i) processing is performed. If the confidence is determined to be high, then group (iii) algorithms can be applied to optimize resources. Different Groups of algorithms (e.g.,

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



Paper 37 at 24-25; Paper 48 at 1-2; EX2036 ¶¶ 48, 86-100; EX1001 28:14-18, 28:1-8

Specification Fully Supports SC 37: Thresholds and a Point Range

- Specification also teaches switching based on a single threshold or a point range

'711 Patent

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

the video frame, as well as a confidence number that is a measure of quality for that prediction. The confidence num-

tion method. If the confidence is below low threshold, falls outside of a range, or the validation method fails, group (i) processing is performed. If the confidence is determined to be high, then group (iii) algorithms can be applied to optimize resources. Different Groups of algorithms (e.g.,

Low Threshold $\equiv$ A Threshold $\equiv$ Activation Number
Range

Measured Confidence Level < A Threshold

Switch to Group (i)

Otherwise

Switch From Group (i) to Group (ii) or (iii) or
Continue with Group (ii) or Group (iii)

Paper 37 at 22, 24; Paper 48 at 1-2; EX2036 ¶¶ 48, 94-95, 146-147, 269; EX2019 ¶¶ 112, 116; EX1001 22:21-29, 27:31-32, 28:14-18



Substitute Claims

Enablement

Speed Adjustment Factor (SAF) / Threshold Limitations Are Enabled

SCs 21, 36, 37	“determining a [] speed adjustment factor [SAF] based on a magnitude of deviation of a direction”
SC 25	“determining a second speed adjustment factor based on the determined distance to the target”
SC 37	“a range defined by [a] first threshold and a second threshold greater than or equal to the first threshold”

- In alleging lack of enablement for SAFs and thresholds, Petitioners conflate
 - Determining **a value** of an SAF / threshold with
 - Determining the **best / optimum value** for the SAF / threshold – which is not claimed

Paper 37 at 7-10, 18, 21, Claims Appendix at 1-5;
EX2036 ¶¶ 145, 114-145; Paper 46 at 2-5; Paper 50 at 3-7

SAF Limitations Are Enabled

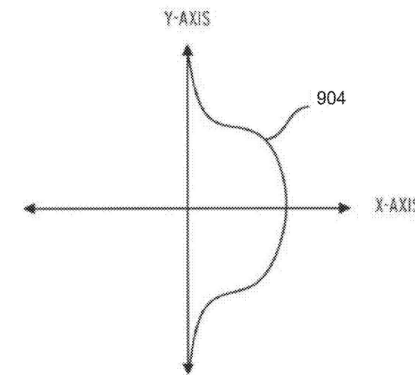
- Specific curves, function constants, are necessary only to determine the best / optimum values

vector. For example, when a **large directional change** is required by the interceptor aerial vehicle to fly towards the target, 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. In some embodiments, the speed and/or accelera-

value. For example, the **distance value determined in 502** 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) and the length of the vector is based on the

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

'711 Patent



$$\text{speed factor}(\text{deg}X) = a / (1 + bc^{-\text{deg}X})$$

Equation 7

$$\text{speed factor}(\text{deg}Y) = a / (1 + bc^{-\text{deg}Y})$$

Equation 8

- With provided guidance, routine/limited experimentation would yield the claimed SAFs

Paper 48 at 3-6; EX2036 ¶¶ 114-129; EX1001 19:8-14, 18:54-58, 22:21-23, FIG. 10

Threshold Limitations Are Enabled

'711 Patent

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. The confidence number can be normalized to a canonical range (0-1) and used to compare the confidence of each different processing scheme to track an item of interest.

Paper 37 at 21-22; Paper 48 at 1, 5-7; EX2036 ¶¶ 76, 78, 80, 141-147; EX1001 27:29-35

NN Parameter (Robustness/Cost/Refresh Rate) Based NN Selection Is Enabled

SC 36	“the first robustness and the first compute cost [of a first NN] being greater, respectively, than the second robustness and the second compute cost [of a second NN]”
SC 37	“the first robustness [of the first NN] being greater than the second robustness [of the second NN] and the first refresh rate [of the first NN] being less than the second refresh rate [of the second NN]”

- Petitioners allege
 - Determination of NN parameters is not disclosed
 - Comparison / selection of NNs based on NN parameters is not enabled

Paper 37 at 18-19, 22-23, Claims Appendix at 4-5
Paper 46 at 5-8; Paper 50 at 7-8

NN Parameters / Selection Are Enabled

- A POSITA would readily know how to make and use claimed systems
- Selection is enabled without knowing NN parameters, by the pre-identified groups

'711 Patent

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+

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.

- At a minimum, a POSITA may simply choose one NN from group (i) and another from group (iii)
- **Selection from provided groups would yield functioning systems per SCs 36 and 37**
- **No undue experimentation would be needed**

Paper 37 at 18-20; Paper 48 at 4-7

EX2036 ¶¶130-140, 153; EX1001 28:40-50, 28:64-29:3

NN Parameters / Selection Are Enabled

- A POSITA would also be able to readily choose from other available NNs
 - From published NN parameters

EX2047 at 7

Tracker	Accuracy	Robustness	EAO
RFL	1.72	3.07	<u>0.2222</u>
DSST	<u>1.95</u>	3.38	0.1814
SiamFC_3s	2.00	2.42	0.2352
SAMF	2.05	<u>2.65</u>	0.1856
KCF	2.20	2.77	0.1924
TGPR	2.73	3.47	0.1811
STRUCK	3.20	4.73	0.1416
MIL	3.68	4.23	0.1645

Table 3. AR ranking and EAO for each tracker.

EX2049 at 9

VI. CONCLUSIONS AND RECOMMENDATIONS

In this paper, a **feedforward neural network** was designed and trained to recognize images of a CrazyFlie 2.0 quadrotor in flight for the purposes of using it in a neighbor sensing system for a swarm of quadrotors. The idea was to be able to give the

backgrounds, or quadrotor orientations. In practice, the neural network is expected to try to identify its neighbors at least a **few hundreds of times per second**. Due to this **high refresh rate**, a 94 percent accuracy is acceptable for the task.

- From such available information, a POSITA would readily select two NNs with little experimentation

Paper 48 at 4-7; EX2036 ¶¶130-140; EX2047 at 7; EX2049 at 9



Substitute Claims

Indefiniteness

SCs 21 and 25 Are Not Indefinite – Specification Describes SAFs

- In the Reply, Petitioners state:
 - “It is unclear from the intrinsic record what [a speed adjustment factor (SAF)] mean[s]”
- In the Sur-reply, Petitioners now argue:
 - This term is unclear because the “specification is not helpful” in explaining “**how** [an] SAF is determined as claimed”
- The Sur-reply argument not an indefiniteness argument; it is a differently worded lack of enablement argument
- Petitioners lack of enablement argument fails, as discussed earlier

Paper 46 at 9; Paper 50 at 9

SCs 21, 25 Are Clear: Specification Provides the Meaning of SAF

At **506**, a vector identifying the relative location of the target is determined using at least the determined relative direction of the target. For example, determined angular utilized to determine the vector. In some embodiments, the vector is at least in part defined using a distance and/or speed value. For example, the distance value determined in **502** is

- A vector
 - Identifies target's relative location
 - Defines distance to the target and speed
- Specified speed is modified using a factor
 - Based on directional change
 - Mapped from distance
 - Scaling
- SAF is a vector element specifying speed modification

'711 Patent

Modifying 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 based on an amount of directional change specified by the vector. For example, when a large directional change is

utilized to determine the vector. In some embodiments, the vector is at least in part defined using a distance and/or speed value. For example, the distance value determined in **502** is mapped to a desired speed of travel to reach the target (e.g.,

At **508**, the vector is scaled. In some embodiments, scaling the vector includes modifying a speed and/or acceleration associated with the vector as well as smoothing and

Paper 37 at 6-8; Paper 48 at 7-8; EX2036 ¶¶ 148-149; EX1001 18:44-46, 18:52-59, 18:65-19:1, 19:4-14

SC 36 Is Not Indefinite – NN Parameters Had Well-Understood Meanings

- In the Reply / Sur-reply, Petitioners state, for the claim terms robustness, compute cost, and refresh rate
 - “It is unclear from the intrinsic record what the terms mean”
 - “Without knowing their boundaries / how they are determined or measured, there is no way of comparing them to determine such a relationship”
 - “[S]pecification does not say **how such a cost is classified** as being a high, medium, or low”
- These are not indefiniteness arguments – they are differently worded lack of enablement arguments
 - Nowhere does SC 36 recite classifying a cost or classifying an NN based on its cost
- Petitioners lack of enablement argument fails, as discussed earlier

'711 Patent

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+

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.

Paper 46 at 9; Paper 50 at 9-10; EX1001 28:40-46, 28:64-29:3

SC 37 Is Not Indefinite – Confidence from Current Image Is Not Used

- Petitioners allege that SC 37 is not clear because it requires an NN to applied to a current image to obtain a confidence level to be used for selecting the NN
- As discussed in the context of sufficiency of support
 - SC 37 recites: “a confidence level **for** a detection of the target in the current image”
 - Not a confidence level **derived from** a detection of the target in the current image
- The claim term is clear and fully supported, as discussed earlier

Paper 46 at 9-10; Paper 48 at 8-9



Obviousness

Asserted Obviousness Combinations

Claim	Asserted Combinations
12	Beard, Laufer Pierce, Laufer Pierce, Kumar, Laufer Pierce, Sahasrabudhe, Laufer Pierce, Kumar, Sahasrabudhe, Laufer
SC 21	Beard, Laufer, Pierce, Kumar, Khan – BLPKK Pierce, Kumar, Khan, Beard – PKKB
SC 36	Beard, Laufer, Pierce, Ning – BLPN Pierce, Kumar, Ning, Beard – PKNB
SC 37	None

Paper 1; 10, 34, 68, 71, 50; Paper 28 at 12-18 at 9-10; Paper 46 at 10-25

Claim 12 Not Obvious – Laufer’s Autopilot Is Remote – Not Autonomous

Petitioners’ Construction	PO’s Construction
“navigated and maneuvered by a computer, without a need for human control or intervention under normal road conditions”	“making navigation decisions by the reference aerial vehicle by-itself, without outside control”

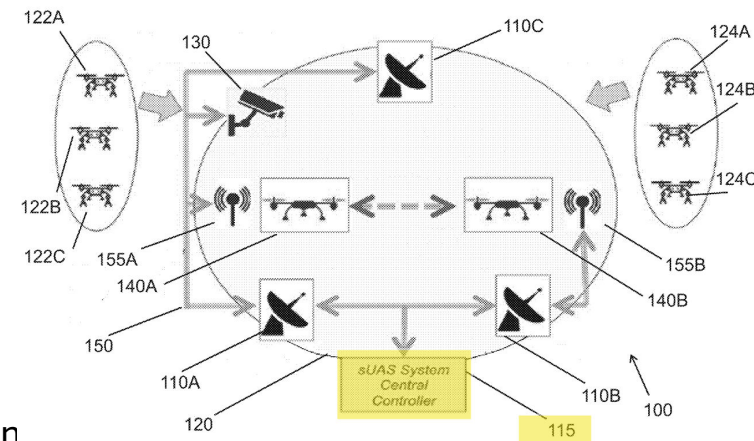
- All asserted combinations rely on Laufer for the “autonomously navigating” limitation

Laufer (EX1010), ¶¶ 40, 30, FIG. 1

[0040] Central controller 115 uses autopilot command module 440 to generate autopilot aircraft commands 445 based on interception location 425 and the interceptor aircraft dynamics model. Central controller 115 uses autopilot

10/100 and Gigabit Ethernet connections. Central controller 115 can be connected to interceptor aircraft 140A-140B through a wireless connection. The wireless connection can

- Laufer’s central controller is a remote controller
- Drone of any of the combination does not navigate auton



Paper 20 at 12-16; EX1010 ¶¶ 30, 40, FIG. 1

SCs Not Obvious: ML Models / NNs Do Not Run on Conventional Processors

- A POSITA would and could not make the BLPKK / PKKB / BLPN / PKNB combinations
 - Petitioners assert that on-board ML model / NNs are implemented by Beard's / Pierce's processors
 - Petitioners allege implementing the ML models / NNs of Laufer, Pierce, Kumar, and Khan on Beard's image processor or Pierce's processor as:
 - 1: Simple programming, and
 - 2: Combining well-known algorithms
 - Both allegations are incorrect

EX2039 at 1

Recent advances in deep learning have made the use of large, deep neural networks with tens of millions of parameters suitable for a number of applications that require real-time processing. The sheer size of these 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. This paper is a tutorial

Paper 1 at 13, 37; EX1003 ¶¶ 114, 155, 161; Paper 46 at 13-17, 21-24, 17-20;
Paper 48 at 10-11; EX2036 ¶¶ 166-170, 179-208; EX2039 at 1

SCs Not Obvious: ML Models / NNs Do Not Run on Conventional Processors

EX2042

- Blue highlight – Signal Processing Algorithm
- Yellow highlight – Neural Networks

TABLE I
HARDWARE SPECIFICATION AND MEASURED PERFORMANCE OF
HAND-CRAFTED FEATURE HOG [7] AND LEARNED FEATURE CNN [8].

Implemented Feature	[7]	[8]	
	HOG	CNN (AlexNet)	CNN (VGG-16)
Technology	65nm	65nm	
Gate Counts (kgates)	893.0	1176.0	
Memory (kB)	159.0	181.5	
Multiplier Bitwidth	5×11 – 22×22	16×16	
Throughput (Mpixels/s)	62.5	1.8	0.04
Throughput (GOPS)	46.0	46.2	21.4
Power (mW)	29.3	278.0	236.0
DRAM access (B/pixel)	1.0	74.7	2128.6
Energy Efficiency (nJ/pixel)	0.5	155.5	6742.9
Energy Efficiency (GOPS/W)	1570.0	166.2	90.7

TABLE II
COMPUTATIONAL COMPLEXITY COMPARISON BETWEEN FEATURES

Feature		GOP/Mpixel	Ratio
Hand-crafted	HOG	0.7	1.0×
Learned	CNN (AlexNet)	25.8	36.9×
	CNN (VGG-16)	610.3	871.9×

Paper 48 at 9-11; EX2036 ¶¶ 54, 166-208; EX2042 at 2

SCs Not Obvious – Substantial Effort Needed for Combinations to Work

- Running ML Models / NNs on Beard's / Pierce's processors would take significant skill / effort
- This was well outside the scope of a POSITA

EX2043 at 2

III. CPU BASED YOLO ARCHITECTURE

Our aim is to build the model (CPU Based YOLO) for CPU Based real time object detection. Developers of YOLO used Darknet [16] framework for running the algorithm. We run YOLOv3 with DarkFlow [17] and OpenCv [10] as only they are favorable for CPU Based YOLO. Our aim was to develop the model in Windows operating system but we found that DarkFlow installation in windows is really a complex task. However, we installed it and run YOLOv3 [12] on it, but the result was terrible. The FPS was too low and the starting time was too much. Then we start to do the task on OpenCv. We input a video though OpenCv and

EX2044

"YAD2K" script to convert the Darknet weights to Keras format, and then write my own script to convert the Keras weights to Metal.

but it does mean we have to change the weights of the preceding convolution layer.

Paper 48 at 9-11; EX2036 ¶¶ 179-193; EX2043 at 2; EX2044 at 7-8

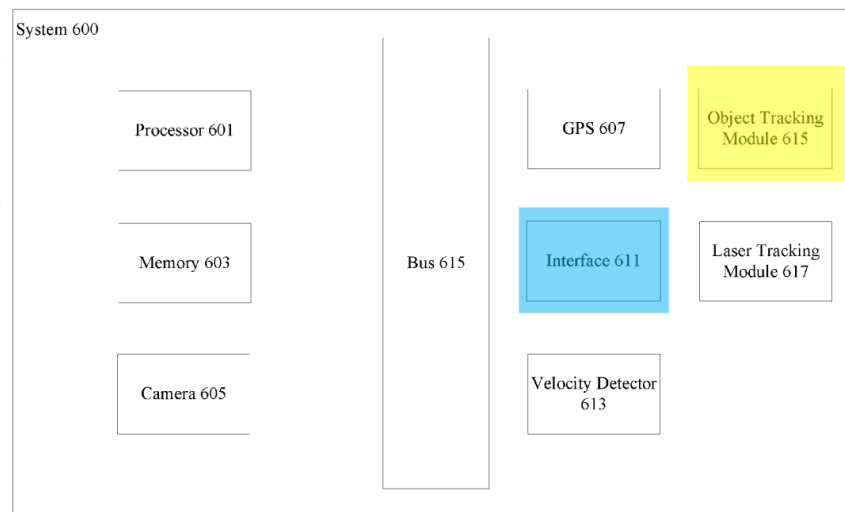
SCs Not Obvious: No NN (Including Pierce's) Is Run On-Board a Drone

[0042] FIG. 6 illustrates one example of a drone system 600 that can be used in conjunction with the techniques and mechanisms of the present disclosure in accordance with one or more embodiments. In various embodiments, drone system 600 may be drone 102 previously described with reference to FIG. 1. However, in other embodiments, various elements of drone system 600 may correspond to separate components, including drone 102, a server, a controller, etc.

Petitioners Sur-reply

to FIG. 1.” EX1005, ¶¶0027, 0042, 0044-0046, Figs. 1, 6. “Drone system 600” may include “object tracking module 615,” and several other on-drone components (camera capturing images in “surrounding area of drone 102,” GPS, velocity detector, laser tracking module). Object tracking module 615 is thus on-board

Pierce (EX1005)



- Module 615 includes a neural network
- In describing system 600, Petitioners omitted its Interface 611, **which cannot be on-board**

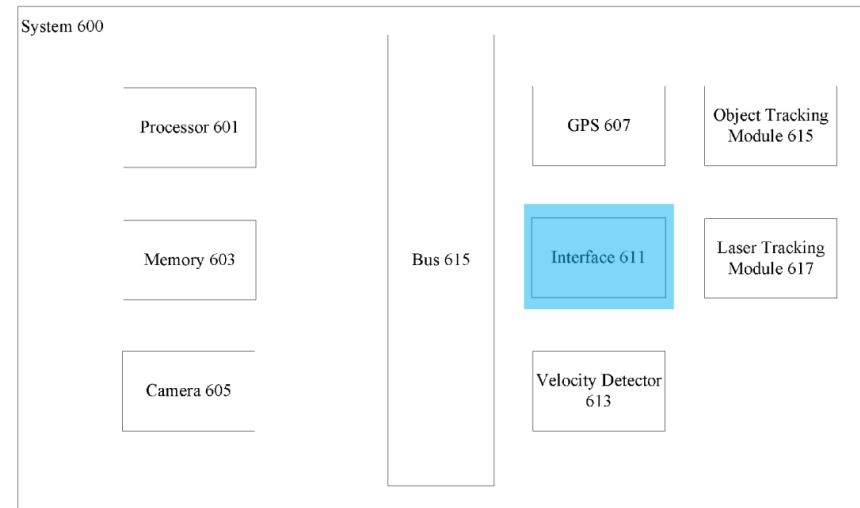
Paper 48 at 11-12; Paper 50 at 11; EX2036 ¶¶ 166-178; EX1005 ¶ 42, FIG. 6

SCs Not Obvious: No NN (Including Pierce's) Is Run On-Board a Drone

Pierce (EX1005)

place of a processor **601** or in addition to processor **601**. The interface **611** is typically configured to send and receive data packets or data segments over a network.

[0043] Particular examples of interfaces supports include Ethernet interfaces, frame relay interfaces, cable interfaces, DSL interfaces, token ring interfaces, and the like. In addition, various very high-speed interfaces may be provided such as fast Ethernet interfaces, Gigabit Ethernet interfaces, ATM interfaces, HSSI interfaces, POS interfaces, FDDI interfaces and the like. Generally, these interfaces may



- Pierce never mentions a wireless interface
- The described interfaces are wired and cannot be on-board a drone

Paper 48 at 11-12; EX2036 ¶¶ 166-178; EX1005 ¶¶ 42-43, FIG. 6

SCs Not Obvious: Pierce NN – a Streaming Server – Cannot Run On-Board a Drone

Pierce (EX1005)

[0038] 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. In some

[0042] FIG. 6 illustrates one example of a drone system 600 that can be used in conjunction with the techniques and mechanisms of the present disclosure in accordance with one or more embodiments. In various embodiments, drone system 600 may be drone 102 previously described with reference to FIG. 1. However, in other embodiments, various elements of drone system 600 may correspond to separate components, including drone 102, a server, a controller, etc.

- Interface 511 of NN system 500 is also a wired interface like interface 611 of drone system 600

Paper 48 at 11-12; EX2036 ¶¶ 166-178; EX1005 ¶¶ 38, 42, FIGS. 5-6

SCs Not Obvious: Cannot Run Different ML Models/NNs on Beard/Pierce Processors

- Even if Pierce's NN system is on-board
- Proposed combination require running different ML models / NNs
 - Each providing **different functionalities**
 - For example, Laufer's classifier, Khan's SVM classifier, or Ning's ROLO, in addition to Pierce's NN
- This is outside the scope of a POSITA
- Not because bodily incorporation is required
- **The functionalities / teachings are not combinable**
- A POSITA would have readily known that ML models / NNs are not plug-and-play
- Providing different functionalities into a single NN requires a full retraining
- Running multiple NNs on a processor compounds the problem discussed earlier

EX2043 at 2

III. CPU BASED YOLO ARCHITECTURE

Our aim is to build the model (CPU Based YOLO) for CPU Based real time object detection. Developers of YOLO used Darknet [16] framework for running the algorithm. We run YOLOv3 with DarkFlow [17] and OpenCv [10] as only they are favorable for CPU Based YOLO. Our aim was to develop the model in Windows operating system but we found that DarkFlow installation in windows is really a complex task. However, we installed it and run YOLOv3 [12] on it, but the result was terrible. The FPS was too low and the starting time was too much. Then we start to do the task on OpenCv. We input a video though OpenCv and

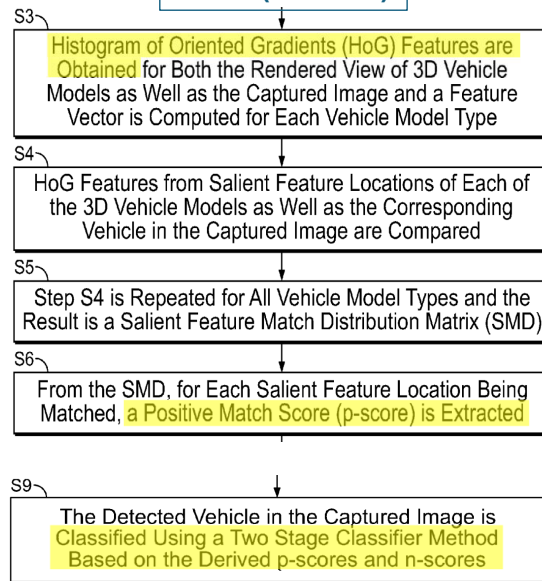
Paper 48 at 9-12; EX2036 ¶¶ 197, 179-202, 207-208; EX2043 at 2

SC21 Not Obvious: Combinations Fails to Teach/Suggest a Limitation

[21.b]

“applying at the reference aerial vehicle 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”

Khan (EX1021)



- Pierce / Kumar
 - Disclose an NN providing a bounding box
 - Petitioners rely on Khan to provide “detected features”
- Khan
 - Silent on bounding box
 - Khan’s HoG (**not an ML model**) detects features
 - Khan’s classifier (SVM – an ML model) does not detect features

Paper 48 at 11-12; EX2036 ¶¶ 216-227; EX1021 FIGS. 3A, 3B

SC21 Not Obvious: Combinations Fails to Teach/Suggest a Limitation

[21.b]

“**applying** at the reference aerial vehicle **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”

- Khan’s features are not the “detected features” recited in [21.b]
- The “detected features” of [21.b] must be identified by an ML model for inclusion in a bounding box
- Khan’s detected features cannot be identified by an ML model for inclusion in a bounding box
 - ML models that provide bounding boxes generate their own features
 - An NN uses its self-generated features (generated in hidden layers) – not externally provided features
 - in generating a bounding box
- **Therefore, none of BLPKK / PKKB teaches / suggests**
 - “applying ... [an ML] model ... to Identify a bounding box ... that includes detected features”

Paper 48 at 11-12; EX2036 ¶¶ 216-227

SC 36 Not Obvious over BLPN and PKNB (No Two NNs / Opposite Costs)

SC 36 ([36.d])	wherein the detecting of the target in the image is performed by applying at the reference aerial vehicle a first neural network at a first robustness and a first compute cost, and
	the determining of the three-dimensional relative location of the target in the image comprises incrementally predicting the location of the target in the image by applying at the reference aerial vehicle a second neural network at a second robustness and a second compute cost,
	the first robustness and the first compute cost being greater, respectively, than the second robustness and the second compute cost.

- Beard and Pierce do not teach / suggest on-board running of NNs
- Ning, upon which Petitioners rely, shows:
 - A single neural network ROLO – not two as alleged
 - Regardless, the cost relationship of Ning’s alleged two NNs is the opposite of that claimed

[Paper 48 at 12; EX2036 ¶¶ 229-233, 256-261, 263-302](#)

SC 36 Not Obvious: Ning's YOLO/LSTM Form A Single NN (ROLO)

Ning (EX1034)

Unlike existing classifier-based methods, the disclosed network can perform object detection and recognition in a **single network evaluation**. As a powerful engine, the dis-

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 lowed and the variables can be set as $S=7$, $B=2$, $C=20$. The **output of this fully connected layer**, is the actual detection result of YOLO. The detections (x, y, w, h, c) at time t is denoted as $L_{t,i}$.

the tracking module. There 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**.

- **The only inputs LSTM receives are from YOLO**

Paper 48 at 12; EX2036 ¶¶ 229-233, 256-261, 263-302; EX1034 5:4-6, 4:55-58, 10:25-37, FIGS. 3, 8

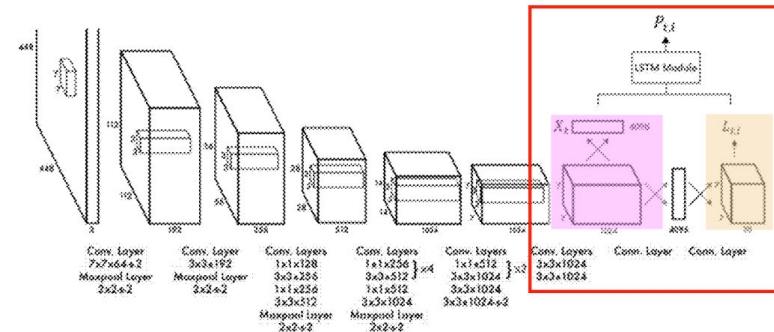


FIG. 8

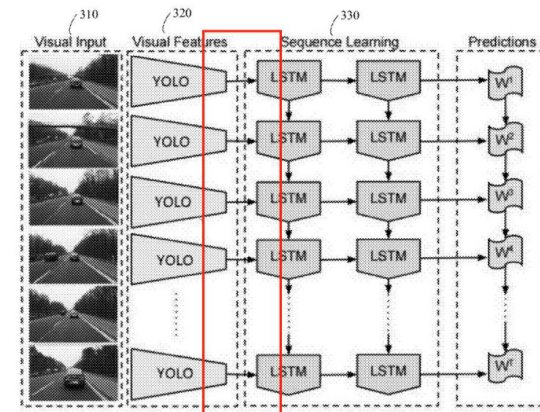
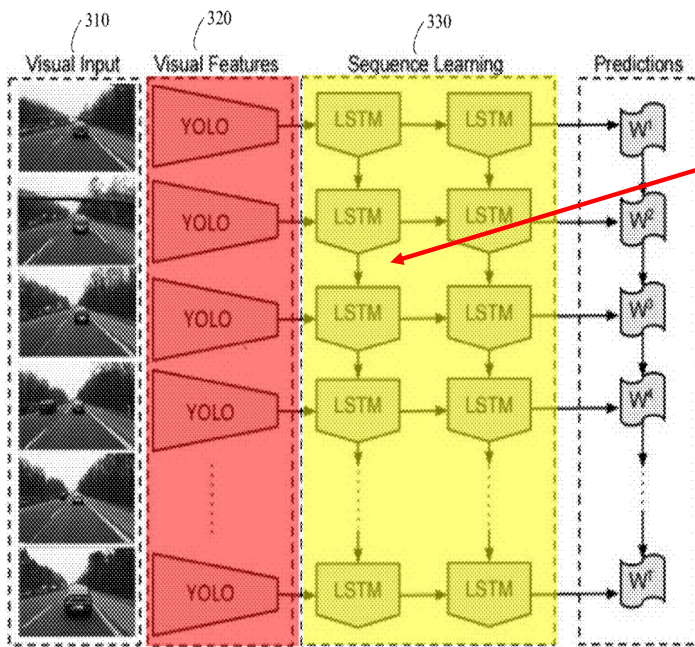


FIG. 3

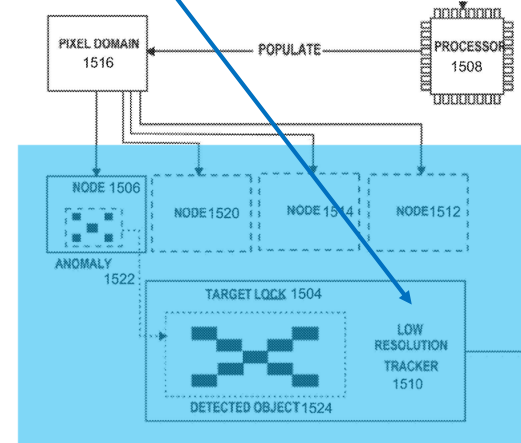
SC 36 Not Obvious: Ning's LSTM (Alleged Second NN) Is Not Applied to an Image

Ning (EX1034)



'711 Patent

domain inconsistencies. 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.



Paper 48 at 12; EX2036 ¶¶ 229-233, 256-261, 263-302;
EX1034 FIG. 3; EX1001 21:67-22:7

SC 36 Not Obvious: Ning's LSTM Cost Is Cumulative of / Greater than YOLO's

- SC 36 requires two NNs having different costs
- The two NNs can have different costs only if they can perform their respective tasks independently
- In Ning's system, the YOLO (alleged first NN) and the LSTM (alleged second NN) cannot do that
 - they are **inextricably coupled**

Ning (EX1034)

the tracking module. There 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.

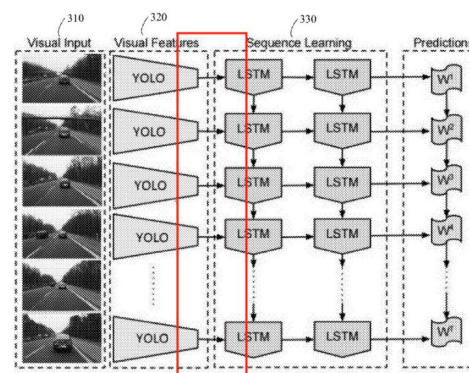


FIG. 3

- LSTM obtains its inputs only from the YOLO, and cannot operate until YOLO has provided those inputs
- The cost (latency) of LSTM is **cumulative** of YOLO's cost
- Cost of LSTM is greater than that of YOLO
- This is the opposite of what is claimed (cost and refresh rate are inversely related)

Paper 48 at 12; EX2036 ¶¶ 229-233, 256-261, 263-302; EX1034 10:34-37, FIG. 3

SC 36 Not Obvious: LSTM's Cost Is Greater than YOLO's – Opposite of Claimed

SC 36

“the first robustness and the **first compute cost [of a first NN] being greater, respectively, than** the second robustness and the **second compute cost [of a second NN]”**

- Regarding opposite cost relationship PO did not incorporate arguments; Reply states, providing pincites:

Paper 48 (PO Reply)

by the YOLO in prior processing the image. This makes **YOLO's cost less than** LSTM's, and not “greater” as [36.d] recites. EX2054 at 7; EX1036 at 48.

EX1036 at 48

*Our unified architecture is extremely fast. Our base **YOLO model processes images in real-time at 45 frames per second.** A smaller version of the network, Fast YOLO,*

EX2054 at 7

Our 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.

- Cost and refresh rate are inversely related
- Dr. Singhose agrees
- “YOLO's cost [is] less” (high refresh rate of 45 fps) “than LSTM's” (low refresh rate of 12 fps)
- Dr. Papanikolopoulos only explained why the results from the two cited references are comparable

Paper 48 at 12; EX2036 ¶¶ 229-233, 256-261, 263-302; EX1036 at 48; EX2054 at 7; EX2033 at 37:6-16

SC 37 Not Obvious

- No prior art based grounds asserted

Paper 46 at 10-11; Paper 48 at 9