Recurrent Motion Neural Network for Low Resolution Drone Detection

Hamish Pratt*, Bernard Evans†, Thomas Rowntree*, Ian Reid* and Steven Wiederman†
* Australian Institute for Machine Learning
The University of Adelaide, South Australia, Australia
Email: hamish.pratt@adelaide.edu.au
† Adelaide Medical School
The University of Adelaide, South Australia, Australia

***Abstract*—Drones are becoming increasingly prevalent in everyday usage with many commercial applications in fields such as construction work and agricultural surveying. Despite their common commercial use, drones have been recently used with malicious intent, such as airline disruptions at Gatwick Airport. With the emerging issue of safety concerns for the public and other airspace users, detecting and monitoring active drones in an area is crucial. This paper introduces a recurrent convolutional neural network (CNN) specifically designed for drone detection. This CNN can detect drones from down-sampled images by exploiting the temporal information of drones in flight and outperforms a state-of-the-art conventional object detector. Due to the lightweight and low resolution nature of this network, it can be mounted on a small processor and run at near real-time speeds.**

***Index Terms*—UAV, Object Detection, Motion**

I. Introduction

Recreational and commercial use of drones has increased substantially in the past few years. These drones can be used for recreational photography or commercial applications such as construction, surveying and animal studies. Drones are affordable, with photography versions selling for approximately $1,000 and light micro-drones for a few hundred dollars. Increasing drone affordability has resulted in their wide-spread accessibility and created the challenge of monitoring active drones. Drones can be flown by individuals from novices to licensed experts. While it is feasible to monitor drones from the ground, this does not permit interception of drones mid-flight. We have identified a requirement for flying drones to use a camera to detect and peruse other drones. Convolutional Neural Networks (CNN) have achieved state-of-the-art accuracy for image classification [1] and object detection [2], [3] which have been used to detect drones in images [4], [5], [6], [7]. Despite their success with drone detection, these conventional object detectors are too large to be mounted on on-board processors and still function in real-time.

Evolutionary processes demonstrate that some hunting insects are able to achieve $\sim$97% capture rates of prey [8] despite low resolution vision. Using a visual system that is tuned to detect small moving targets [9], [10], insects such as dragonflies are able to catch their prey with limited visual acuity and low power, computationally efficient processing. We introduce a novel neural network architecture for detecting small, low resolution drones. Inspired by the low spatial resolution and motion-based vision of dragonflies, we further introduce a recurrent module into our neural network to improve the detection of small moving drones.

Previous neural network architectures have integrated recurrent memory modules for object detection [11], [12], [13]. Our proposed network utilises recurrent modules in a specialised network to detect drones based on temporal information from the scene. Appearance-based detection of flying drones in cluttered environments can be difficult due to their high speeds and small size. Instead, it may be the motion signature of a drone that reveals its location, especially in the case of low resolution images. We analyse the performance of a recurrent neural network in scenarios where motion signatures are crucial for detection and show that our recurrent neural network can outperform an equally sized single-frame detector for small object detection and a state-of-the-art object detector by utilising the temporal information of the scene. We further explore the conditions where our recurrent motion network has the greatest performance increase and demonstrate its viability as a mounted drone detector.

II. Related Work

A. Traditional Object Detection

In recent years, major developments have been made towards detecting objects based on single frame images with neural networks [14], [15], [16] , [17], [2], [3]. These convolutional neural networks follow two main approaches. They can use region proposals to determine areas where objects may be present and pass these areas through an object classifier [14] , [15], [3]. Alternatively they can use a single network to locate and classify the objects in a single pass [16], [17], [2]. While these approaches have mostly enabled single frame object detection, the models are too large and computationally expensive for on-board processing on a drone.

B. Video Object Detection

Object detectors for videos need to process a larger number of sequential frames whilst also handling distorted images due to motion blur or unclear poses. These video object detectors must process frames rapidly whilst also maintaining accuracy. A key component to achieving one or both of these

requirements is to utilise the temporal information between frames of a video. Optical flow, the movement of pixels between frames, can be used to reduce the complexity of detection, as it is cheaper to calculate than doing object detection in images. Approaches have warped a feature map of a CNN object detector with the optical flow [18], [19], [20], or other motion information [21], between the current frame and a future frame to create a prediction of object locations. The optical flow approach reduces the computational cost of object detection because the traditional object detector sparsely performs its expensive computations. Through a lightweight architecture, it is possible to run these networks on small devices such as mobile phones [19]. This lightweight future prediction via motion information greatly improved the speed in exchange for accuracy, but the reverse is also possible. A central key frame can be selected, and the feature maps of frames before and after it in a video sequence can be warped via optical flow to the key frame. The collection of these frames can be aggregated to produce a more robust prediction of objects in the scene [22]. Instead of manipulating a scene through optical flow, other approaches have exploited recurrent Long Sort Term Memory Modules (LSTMs) to propagate the temporal information through the CNN [12]. By using lightweight convolutional LSTMs, fast video object detectors have been implemented for mobile phones [13].

C. Drone Detection

Recently, CNN object detection methods have been applied to drone detection instead of everyday objects. Several of these methods take advantage of transfer learning of conventional object detectors to specialise them towards drones [4], [5], [6], [7]. Such approaches use computationally expensive Region Proposal Networks (RPN) [5], [6] , [7]. Alternatively, other approaches have utilised a modified You Only Look Once (YOLO) architecture [4]. One unique approach to drone detection uses a small sliding window across the image to detect areas of interest [23]. Small cut-outs of the image are stacked together along the length of the video, stabilised to keep the object centred and then classified if they are objects of interest. Through stacking multiple frames, the network is utilising temporal information, but through stabilisation, any movement across the scene is undefined. Despite its success, the slow sliding window approach is impractical for rapid detection on computationally constrained hardware. Another temporally focussed drone detector evidenced that motion characteristics in a network can perform better at detecting small objects as opposed to a single frame detector [24]. Their network sent a portion of an image through a convolutional LSTM to help determine if an object, such as a bird, was moving. This technique required a template of an image to follow, determining the initial template via image subtraction if the camera was stationary, or through a sliding window approach [23] when the camera was in motion. As our neural network aims to be implemented on a moving platform and requires quick detection speeds, neither an image subtraction nor sliding window method is practical.

D. WAMI

Wide field motion imagery (WAMI) is a technique used to identify small moving objects in cluttered environments. WAMI differs from normal object detection methods as detections are based on single pixel locations and not bounding boxes. WAMI images can contain thousands of small objects in cluttered environments and significantly larger image sizes [25]. Many WAMI techniques use image subtraction to detect moving objects in the scene [26] which becomes an issue when the platform has its own ego motion. Pure appearance based detectors would have difficulty detecting objects due to numerous small and blurry objects. Recently, temporally aware networks have been developed by processing multiple frames at the same time to capture motion [27], [25] with one approach using a two stream appearance-motion network [25]. A 3D CNN takes in temporal stacks of input to generate the region proposals which are then passed to smaller specialised detectors like other RPN object detectors.

III. Networks

Our task involves single class detection of small objects - two things that a standard YOLO architecture is not built for. We further present the additional challenge of detection in a low resolution image. To overcome the issue of small object detection, multiple scale detection was introduced [2] which involved more convolutions. YOLO is also a network designed to differentiate between multiple object classes in a scene, whereas we only need to detect a single class of interest. Therefore layers of a network which are designed to capture feature information of a broad range of objects are likely superfluous. A standard YOLO object detector does not perform well in the challenge of drone detection (Fig. 8).

Our motion network (Fig. 1) is custom designed for the task of drone detection. The network is considerably smaller in size to YOLO (Table. III) and we can avoid unnecessary class comparisons. By utilising upsampling and skip layers in the network, we are able to combine early semantic information in the low information space. We expect this to be useful for the preservation of small objects and prevent their information from vanishing through downsampling. Our network was also inspired by FlowNet [28] which was able to determine motion information (i.e. optical flow) between two frames. Two Convolutional Gated Recurrent Unit (ConvGru) [29] blocks were added into our network as a means to transfer temporal information across frames. Each ConvGru block consisted of three linked ConvGru cells. The aim of this network is to understand aberrant motion of the scene through temporal recurrence, but also to distinguish an object through appearance.

A single frame model comparison was created to compare the performance of the motion network. This network was made equal in size to establish a fair performance comparison. We replaced each individual ConvGru Cell with three convolutional blocks to mimic a ConvGru cell. Both models received a 3*128*128 tensor and outputted a 5*15*15 tensor similar to a single shot detector for object detection. The five

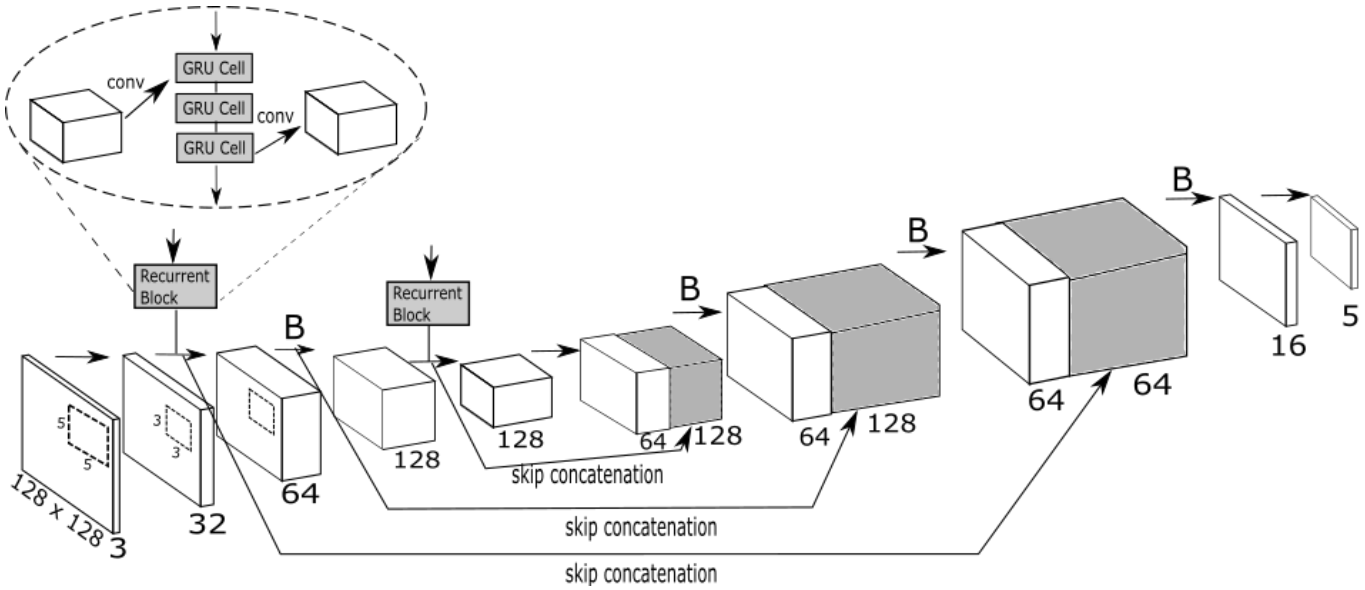


Fig. 1: The recurrent motion network takes in a 3 channel 128*128 image and outputs a final 5 channel 15*15 tensor that predicts drone location and bounding box dimensions. Transitions with a "B" above mark bottle-neck layers. Each convolutional block in the network (besides the last) consists of batch normalisation and a Leaky RELU activation layer.

tensors denoted the confidence, x-offset, y-offset, bounding box width and bounding box height for a drone in each cell.

IV. Experiments

Our primary aim was to determine if integrating motion information could improve the detection of drones in the dataset. We also addressed if the network was understanding motion - and classified the situations where the motion network outperformed the single frame network. The drone dataset consisted of YouTube videos of drones in flight, self-recorded drones in flight, and existing annotated drones in datasets [30], [23]. The videos of drones (both RGB and greyscale) ranged from first person view chase scenes, recordings of drones from the ground, and recordings of drones in flight from an unstable drone platform. Our dataset (29*k* images for test and 66*k* images for train) is considerably more difficult than some previously used images of drones and aims to replicate the challenge of searching for drones in a noisy environment. Human annotators would often require temporal information to locate the drones and many drones were in cluttered environments. Images were 720*720 random crops (or largest possible) of the original image, scaled down to a 128*128 image. Bounding boxes were constrained to a minimum size of 7*7 pixels to prevent the down sampled bounding box misrepresenting the extent of the drone. The dataset presents the challenge of drones with ego-motion chasing other drones where image subtraction may not be applicable. In these scenarios, the target drone is stationary whilst the background is moving. Here the network must differentiate coherent scene motion by extracting the objects moving differently (in the low resolution image). In the contrary scenario, some videos are recorded from a drone almost stationary while drones move across the field of view. Therefore the motion network ideally must understand both conditions and look for aberrant motion. The motion model was trained on 15 frame sequential chunks to learn the temporal information. The 15 frames had a consistent random crop and random flip throughout each video sequence. The loss was calculated as the average of the loss over all individual frames with a small decay added to the earlier frames to promote learning the motion of the scene. The loss of a sequence can be equated as:

$$loss = \frac{\sum_{15}^{i} 0.9^{(15-i)} * frame\_loss(frame_i)}{15} \quad (1)$$

The frame loss for each individual frame was the mean square error loss from the model output and the ground truth location and bounding boxes. The single frame model was trained on the same dataset but treated each frame as independent for the purpose of calculating the loss. All models were trained for a maximum 500 epochs at a learning rate of 0.0003 that decreased by 0.85 every 50 epochs. The specific model weights were chosen by the iteration with the best performance on a validation set (a variant of the test set). Performance was evaluated using the precision-recall metric for the bounding boxes. Non-maximum suppression reduced any boxes overlapping greater than 0.4 intersection over union (IOU). Successful detections were recorded with an overlap threshold of 0.4 IOU. If multiple bounding boxes existed and each successfully detected the same drone, these multiple successful detections would not increase the recall beyond the first but would not decrease precision.

A. Is Motion Useful?

The aim for the ConvGrus and the recurrent network is to understand the motion of the scene and use it to detect

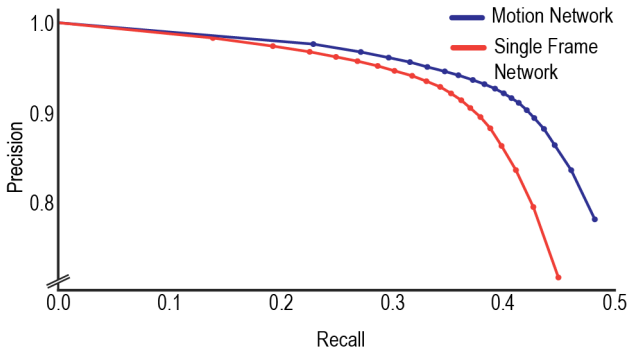


Fig. 2: Average Precision-Recall of the motion network and single frame network on the custom drone dataset.

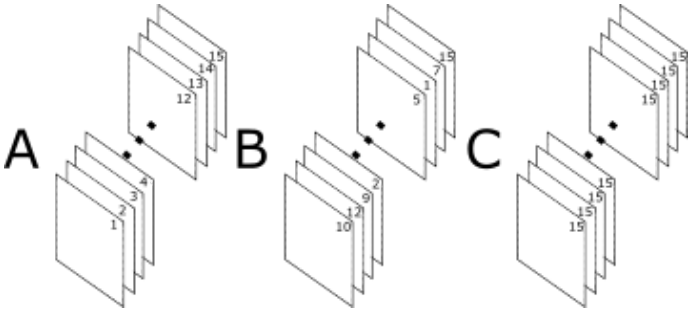


Fig. 3: The three subsets of the 15 frame chunks. (A) is all 15 frames in order, (B) is randomised with the 15th frame in order, and (C) is the last frame repeated.

drones. If the proposed motion network is able to understand and process additional information through motion, it would be expected to outperform the single frame network. To test this, we recorded the precision and recall for detections on each frame for both networks. Detection recording started after 14 frames of each scene to ensure the motion network had time to build up temporal information.

The motion network displayed improved performance over the single frame network on the drone dataset (Fig. 2) - especially at lower confidence thresholds. We believe that this improved accuracy can be attributed to the network's ability to understand and process the motion in the scene. However, it is possible that the difference could be due to other properties of the ConvGru (i.e. not temporal information)

B. Does the Network Use Motion?

If the motion network is not utilising the motion of the scene and instead just collates temporal information, we would expect that the motion network would perform analogously without ordered scene information. For this test we divided the dataset up into unique continuous 15 frame chunks modified into three subsets (Fig. 3):

1) All 15 frames passed in order
2) The 14 initial frames randomly shuffled in a random order with the last frame in correct place
3) The final frame repeated 15 times

A single frame comparison was passed the same dataset which saw only the final 15th frame. The precision-recall metric for all variations was calculated based on the 15th frame (Fig. 4). These results demonstrate that the motion network is doing more than combining static information over time. When it observed the same frame 15 times, it performed the poorest. This indicates that not only is history required, but that the motion network requires that history to be coherent, (i.e. representative of continuous motion).

Having determined that the motion network uses continuous temporal information to improve detections, we tested how much of this information is useful. The motion network was trained on 15 frame chunks, but it is unclear if it achieves a maximum memory of temporal of information before, at or

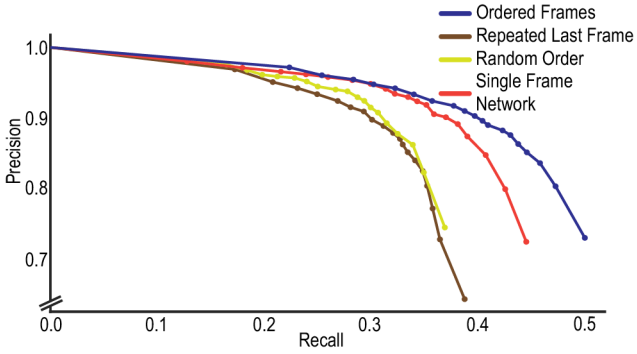


Fig. 4: Average Precision-Recall of the motion network being passed varying formats of 15 frame chunks. Detection calculations were only performed on the final frame. Ordered Frames is the standard input to the motion network as presented above. The Repeated Last Frame model simply repeated the detection frame 15 times (i.e. no real history). The Random Order model shuffled the order of the fourteen frames used as history.

after 15 frames. We split the dataset up into chunks of 25 frames, and then tested its detections on the final frame. A priming value determined how many real frames were shown before the final frame - with the remaining prior frames passed as black frames. If the network is shown 3 priming frames it would be shown 21 black frames instead of the 21 actual frames, and then passed the next real 3 frames in the sequence. The area under the precision-recall curve was calculated for several priming values with the precision-recall metric determined on detections from the final 25th frame (which was never a black frame). Despite the network being trained on 15 frames, it achieves maximum performance beforehand at around 10 frames. (Fig 5). There is a big difference between showing the network no temporal information (0 priming), and minimum temporal information (1 frame priming). These results suggest that the network requires temporal information to improve detection accuracy and that its memory does have a distinct limit.

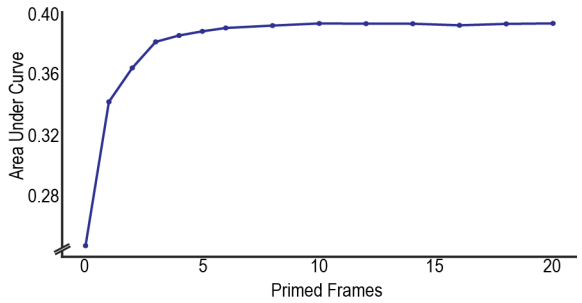


Fig. 5: Precision-Recall of the motion network as the number of temporal priming increases.

TABLE I: Object size categories.

Bounding Box Side Length (Pixels)	# of Ground Truth Drones
0-8	14836
8-16	2568
16-32	302
32-64	177

C. How is Motion Useful

The motion network uses continuous temporal information from previous frames to improve its detection accuracy. It is unclear, however, under what conditions the motion network is able to detect the drone where the single frame network cannot. We examined the dataset based on object size and speed to determine situations where the motion network had improved detection rates. A confidence threshold of 0.05 was selected to gather the largest number of scenarios.

1) Object Size: If the recurrent motion network is utilising motion, we would expect that smaller objects are easier to see as they would cause temporal flicker, even if they cannot be seen by their appearance. To test this hypothesis, we split the dataset up based on the average side length of the ground truth bounding box into 4 categories (Table. I). The results (Fig. 6) support the hypothesis that the motion network is better at detecting small objects, but it highlights that it is worse at detecting larger objects. While a deeper network like the single-frame network is more beneficial for identifying the shape of larger drones, temporal information is crucial for the smaller object detection.

2) Object Speed: To examine the effect of object speed on drone detection, we split the video chunks up into frames where only one drone was present. This was to avoid any object association issues of multiple drones in view. For two consecutive frames with a single drone in each, the distance travelled by the centre-point of the drone was calculated (Table. II) and then recorded if the drone was detected (Fig 7a). The distance travelled was split up into four categories depending on the percentage of the 128*128 image travelled.

Since the recurrent network was ideally detecting motion, we would expect that it would improve detections when the drone was moving across the screen. The network has its best detection in the cases of low motion. It should be noted that in "0%" travelled case, it is possible that the drone

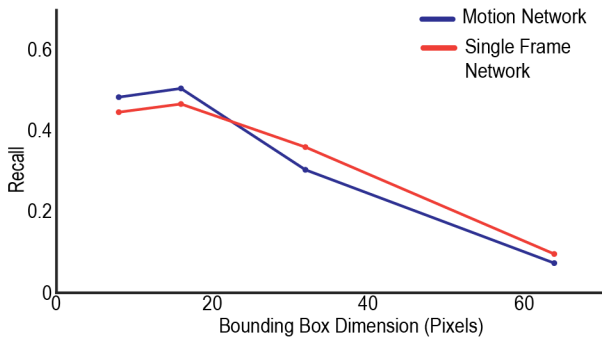
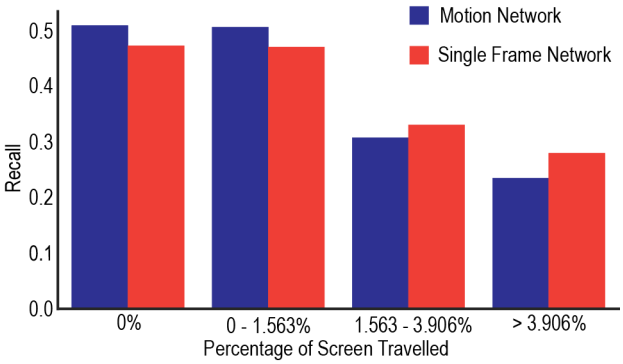
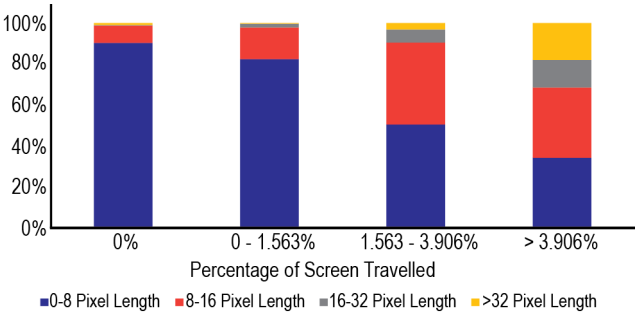


Fig. 6: Recall for varying object sizes in the custom drone dataset.



(a) Recall for varying object speeds in the custom drone dataset.



(b) Object sizes of drones across the four different speeds

Fig. 7: Detection for different speeds

TABLE II: Object speed categories.

Percentage of Image Travelled	# of Ground Truth Drones
0%	7060
0-1.563%	8076
1.563-3.906%	1178
>3.906%	178

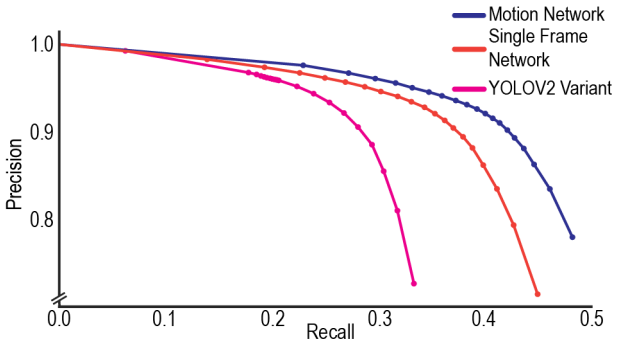


Fig. 8: Precision-Recall comparison of the single frame and motion network when compared against a YOLOV2 variant trained on the same dataset.

is still moving but instead the camera platform is moving forward due to a chase scene. These results suggest that the motion network performs best when it is easy to correlate the drone's previous position and its new position. In the situations where the drone travelled many pixels, the single frame appearance network performed better. In this scenario, it would be harder to correlate the drones' positions together and needs to be solved entirely with appearance cues. By comparing the distribution of object sizes across the four speeds, we do find that the smaller objects move less while the larger drones move further across the screen (Fig. 7b) which may also explain the improved detection rate of the small objects.

D. State-of-the-Art Comparison

We have observed the improvement in the precision-recall performance of the motion network compared to a single frame network. However, it is unclear how the network compares against a conventional object detector in speed and performance. We recreated the YOLOV2 variation used by Aker and Kalkan for drone detection [4] and modified it to process 128*128 and return the same output format. The YOLOV2 variant was trained from scratch using the custom drone dataset. Both our networks achieved a greater precision-recall than the YOLOV2 variant (Fig. 8) and these results demonstrate that a conventional YOLO network is not suited to the task of single class small object detection with low resolution input. The single frame network out performing YOLO suggests that the network structure of preserving early layer information is vital to detection.

E. Hardware Deployment and Network Size

With the aim that these networks could be deployed on a small onboard processor and mounted on a drone, we analysed the speed performance of all the algorithms on a Jetson Nano. The Jetson Nano was connected to a Logitech webcam which passed a live stream to the networks to determine their speeds (Table. III).

Both our single frame and motion networks have considerably less parameters (Table. III) and run considerably faster than the conventional YOLO network. Even with a larger network capable of more learning, the standard YOLO architecture is still outperformed for the single object detection task. Our networks achieve near real-time speeds on the Jetson Nano and are favourable for deployment.

TABLE III: Network Characteristics

Network	Jetson Nano FPS	# Parameters
Motion Network	20	$4.1 * 10^6$
Single Frame Network	24	$2.5 * 10^6$
YOLOV2 Variant	6	$6.7 * 10^7$

V. Conclusion

We have introduced a CNN architecture specialised for detecting drones and explore the addition of a temporal module. Our architecture aims to preserve small object appearance and transfer it to the lower information space. Small moving drones are hard to identify without additional information and through the implemented recurrent modules, the motion network can detect objects that a single frame network cannot. While a deeper network is beneficial in detecting larger objects, there is a benefit to utilising temporal information where appearance is limited. Our networks demonstrate the unsuitability of a conventional YOLO object detector on single-class small object detection - especially in scenarios where performance is essential. With the low computational requirement of our motion network, it can achieve near real-time results when mounted on a Jetson Nano. Our results can lead to more domain specialised neural network architectures and for closed system object detection.

VI. Acknowledgements

This work was supported by Australian Research Council funding schemes; Future Fellowship (FF180100466), Laureate Fellowship (FL130100102) and the Centre of Excellence for Robotic Vision (CE140100016).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv:1804.02767 [cs]*, Apr. 2018, arXiv: 1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767

[3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015. [Online]. Available: http://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks

[4] C. Aker and S. Kalkan, "Using Deep Networks for Drone Detection," *arXiv:1706.05726 [cs]*, Jun. 2017, arXiv: 1706.05726. [Online]. Available: http://arxiv.org/abs/1706.05726

[5] M. Nalamati, A. Kapoor, M. Saqib, N. Sharma, and M. Blumenstein, "Drone Detection in Long-Range Surveillance Videos," in *2019 16th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Taipei, Taiwan: IEEE, Sep. 2019, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8909830/

[6] M. Saqib, S. Daud Khan, N. Sharma, and M. Blumenstein, "A study on detecting drones using deep convolutional neural networks," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Lecce, Italy: IEEE, Aug. 2017, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/document/8078541/

[7] A. Schumann, L. Sommer, J. Klatte, T. Schuchert, and J. Beyerer, "Deep cross-domain flying object classification for robust UAV detection," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. Lecce, Italy: IEEE, Aug. 2017, pp. 1–6. [Online]. Available: http://ieeexplore.ieee.org/document/8078558/

[8] R. M. Olberg, A. H. Worthington, and K. R. Venator, "Prey pursuit and interception in dragonflies," *Journal of Comparative Physiology A: Sensory, Neural, and Behavioral Physiology*, vol. 186, no. 2, pp. 155–162, Feb. 2000. [Online]. Available: http://link.springer.com/10.1007/s003590050015

[9] S. D. Wiederman, J. M. Fabian, J. R. Dunbier, and D. C. O'Carroll, "A predictive focus of gain modulation encodes target trajectories in insect vision," *eLife*, vol. 6, p. e26478, jul 2017. [Online]. Available: https://doi.org/10.7554/eLife.26478

[10] B. H. Lancer, B. J. Evans, J. M. Fabian, D. C. O'Carroll, and S. D. Wiederman, "A Target-Detecting Visual Neuron in the Dragonfly Locks on to Selectively Attended Targets," *The Journal of Neuroscience*, vol. 39, no. 43, pp. 8497–8509, Oct. 2019. [Online]. Available: http://www.jneurosci.org/lookup/doi/10.1523/JNEUROSCI.1431-19.2019

[11] X. Chen, J. Yu, and Z. Wu, "Temporally Identity-Aware SSD With Attentional LSTM," *IEEE Transactions on Cybernetics*, pp. 1–13, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8638831/

[12] G. Ning, Z. Zhang, C. Huang, Z. He, X. Ren, and H. Wang, "Spatially Supervised Recurrent Convolutional Neural Networks for Visual Object Tracking," *arXiv:1607.05781 [cs]*, Jul. 2016, arXiv: 1607.05781. [Online]. Available: http://arxiv.org/abs/1607.05781

[13] M. Zhu and M. Liu, "Mobile Video Object Detection with Temporally-Aware Feature Maps," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 5686–5695. [Online]. Available: https://ieeexplore.ieee.org/document/8578694/

[14] R. Girshick, "Fast R-CNN," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago, Chile: IEEE, Dec. 2015, pp. 1440–1448. [Online]. Available: http://ieeexplore.ieee.org/document/7410526/

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," *arXiv:1311.2524 [cs]*, Nov. 2013, arXiv: 1311.2524. [Online]. Available: http://arxiv.org/abs/1311.2524

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 779–788. [Online]. Available: http://ieeexplore.ieee.org/document/7780460/

[17] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI: IEEE, Jul. 2017, pp. 6517–6525. [Online]. Available: http://ieeexplore.ieee.org/document/8100173/

[18] X. Zhu, J. Dai, L. Yuan, and Y. Wei, "Towards High Performance Video Object Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, Jun. 2018, pp. 7210–7218. [Online]. Available: https://ieeexplore.ieee.org/document/8578851/

[19] X. Zhu, J. Dai, X. Zhu, Y. Wei, and L. Yuan, "Towards High Performance Video Object Detection for Mobiles," *arXiv:1804.05830 [cs]*, Apr. 2018, arXiv: 1804.05830. [Online]. Available: http://arxiv.org/abs/1804.05830

[20] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, "Deep Feature Flow for Video Recognition," *arXiv:1611.07715 [cs]*, Nov. 2016, arXiv: 1611.07715. [Online]. Available: http://arxiv.org/abs/1611.07715

[21] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. C. Loy, and D. Lin, "Optimizing Video Object Detection via a Scale-Time Lattice," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 7814–7823. [Online]. Available: https://ieeexplore.ieee.org/document/8578913/

[22] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-Guided Feature Aggregation for Video Object Detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*. Venice: IEEE, Oct. 2017, pp. 408–417. [Online]. Available: http://ieeexplore.ieee.org/document/8237314/

[23] A. Rozantsev, V. Lepetit, and P. Fua, "Detecting Flying Objects Using a Single Moving Camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 5, pp. 879–892, May 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7466125/

[24] R. Yoshihashi, T. T. Trinh, R. Kawakami, S. You, M. Iida, and T. Naemura, "Differentiating Objects by Motion: Joint Detection and Tracking of Small Flying Objects," *arXiv:1709.04666 [cs]*, Sep. 2017, arXiv: 1709.04666. [Online]. Available: http://arxiv.org/abs/1709.04666

[25] R. LaLonde, D. Zhang, and M. Shah, "ClusterNet: Detecting Small Objects in Large Scenes by Exploiting Spatio-Temporal Information," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT, USA: IEEE, Jun. 2018, pp. 4003–4012. [Online]. Available: https://ieeexplore.ieee.org/document/8578519/

[26] L. W. Sommer, M. Teutsch, T. Schuchert, and J. Beyerer, "A survey on moving object detection for wide area motion imagery," in *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Lake Placid, NY: IEEE, Mar. 2016, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/document/7477573/

[27] C. Hartung, R. Spraul, and W. Krüger, "Improvement of persistent tracking in wide area motion imagery by CNN-based motion detections," in *Image and Signal Processing for Remote Sensing XXIV*, L. Bruzzone, F. Bovolo, and J. A. Benediktsson, Eds. Berlin, Germany: SPIE, Oct. 2018, p. 26. [Online]. Available: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10789/2325367/Improvement-of-persistent-tracking-in-wide-area-motion-imagery-by/10.1117/12.2325367.full

[28] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning Optical Flow with Convolutional Networks," in *2015 IEEE International Conference on Computer Vision (ICCV)*. Santiago: IEEE, Dec. 2015, pp. 2758–2766. [Online]. Available: http://ieeexplore.ieee.org/document/7410673/

[29] J. Kimmel, "Convolutional gated recurrent unit (convgru) in pytorch," 2017. [Online]. Available: https://github.com/jacobkimmel/pytorch_convgru

[30] H. K. Galoogahi, A. Fagg, C. Huang, D. Ramanan, and S. Lucey, "Need for Speed: A Benchmark for Higher Frame Rate Object Tracking," *arXiv:1703.05884 [cs]*, Mar. 2017, arXiv: 1703.05884. [Online]. Available: http://arxiv.org/abs/1703.05884