JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 39, 549-560 (2023)
DOI: 10.6688/JISE.202305_39(3).0006

A Multi Tasking Model for Object Detection, Instance Segmentation and Keypoint Estimation Tasks*

AJAI JOHN CHEMMANAM[1], BIJOY A JOSE[2,+] AND ASIF MOOPAN[3]
[1]*Cyber Physical Systems Lab, Department of Electronics*
[2]*Cyber Physical Systems Lab, Department of Computer Science*
Cochin University of Science and Technology
Kerala, 682022 India
E-mail: {ajaichemmanam; bijoyjose}@cusat.ac.in
[3]*Vuelogix Technologies Pvt. Ltd.*
Kerala, 682037 India
E-mail: asif@vuelogix.com

Recent developments in neural networks have enabled them to achieve comparable or better than human accuracy on different computer vision tasks like image classification, object detection, segmentation, keypoint estimation *etc*. Large scale datasets have explicitly been curated for each of these tasks, and researchers from around the world compete to get state of the art results on these benchmarks. Outside these datasets, these models may not always perform equally better. Real-world applications often require multiple models to be used together to generate meaningful results. The processing time primarily relies on the slowest model in the pipeline. The allocated resources for other models remain idle in the meantime, causing a sub-optimal processing pipeline. Each of these models comes with its own input data pre-processing, feature extraction, output post-processing *etc*., causing a significant unnecessary overhead.

To overcome these issues, we explored multitasking architectures to do multiple closely related tasks together. In this work, we developed a single multitasking model to perform object detection, instance segmentation and keypoint estimation tasks. We presume such models will be more robust to data specific noises as it finds a better representation of the trained data by learning to predict multiple closely related tasks. Our most accurate model gave 41.2 AP on object detection, 38.2 AP on instance segmentation and 53.0 AP on keypoint estimation tasks when evaluated on COCO validation dataset. We optimised the models through layer fusion and float 16 quantisation. We achieved 107 frames per second (fps), while a lighter version achieved 131fps on RTX 3090 GPU. We also benchmarked the models on Nvidia Jetson Tx2 and got 4.2 fps and 6.3 fps for the two respective models. The models were successfully deployed in a server-client system architecture / cloud computing with future possibilities for on-premise deployment using edge devices.

Keywords: joint learning, multi tasking, object detection, instance segmentation, keypoint estimation

Received Februay 5, 2022; revised April 3, 2022; accepted May 27, 2022.
Communicated by Jimson Mathew.
* This research work was funded and supported by Vuelogix Technologies Pvt Ltd, Confederation of Indian Industry (CII) and the Department of Science and Technology (DST-SERB), Government of India, through the Prime Minister's Fellowship for Doctoral Research 2020.
+ Corresponding author.

1. INTRODUCTION

Computer vision problems such as image classification, object detection, image segmentation *etc*. have been solved at par or better than human accuracy levels. Industries have started using these algorithms in mainstream data analytics to generate new business insights. Traditionally they have been using signal processing and image processing techniques to do image and video analytics but were far from being useful for practical usage. Researchers have developed various models to solve each of these problems with varying model complexity, performance and accuracy. They all are state of the art in one way or another on the specific data they are trained on. As these algorithms are scaled from proof of concepts to real-world problem solvers, they start to degrade their performance when given a different data distribution than they are trained with.

Our industry partner, Vuelogix Technologies, works in the surveillance and security domain. They provide advanced security solutions through AI and ML. Their business requirements primarily include detecting people, helmets, weapons and other objects. They also wanted to do body pose estimation and identify their actions to provide alerts proactively to possible security threats. Anomalous activities like loitering are detected by tracking the people continuously. These require additional models for keypoint estimation, pose classification, tracking algorithms, instance segmentation models to separate objects from the background to improve tracking, *etc*..

The processing power and computation time required for each model are not identical, resulting in an unoptimised processing pipeline and wastage of resources. The processing time of the system largely depended on the slowest model. Running models parallely is not a viable solution as it increases the peak computation power and GPU memory requirement. Also certain models can only be run sequentially. For example in the case of top-down keypoint estimation models, an object detection model must first detect a person in an image. Once we get the results, the bounding box is cropped to each person and then given for body keypoint estimation by the second model. Further each model has its own data preprocessing and output post-processing techniques. Even though the models can be GPU accelerated, their pre and post-processing are mostly done on CPUs. In most cases, these processing overheads take equivalent or more time than actual model inference. Also, each model has its own feature extraction layers that extract relevant information for each specific task. The features extracted by one model are not reused by another model, leading to significantly more computations.

These challenges quickly overwhelmed the system, which has to provide real-time alerts using the limited computation resources. In order to overcome these issues, we explored the possibility of using multitasking architectures to reduce the number of models and improve the processing pipeline. Multitasking models learn to perform multiple related tasks at the same time, there by reducing the number of models to be used in the pipeline. In addition, the correlation between the tasks could boost the overall performance of the model. The prior knowledge on a related task can help the system learn faster and, at the same time, be more generalised and robust model.

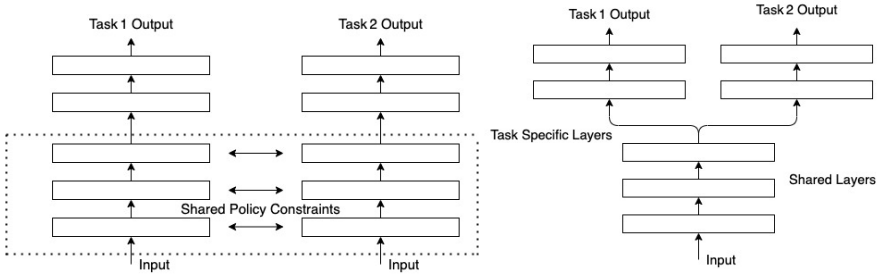
The following sections discuss the different multi-tasking model architectures and the related literature reviews. We detail the proposed model architecture, objective/loss function, training, and inference methodology. Part of this work was originally presented in ICDSE 2021 [1]. We further extended the work by optimising the performance and

evaluating on different hardware configurations for edge and cloud processing.

2. RELATED WORKS

Multitask learning solves multiple related tasks together. It is also known as joint learning, learning to learn, learning with auxiliary tasks *etc*. In the following sections, we discuss more about multitasking architectures and the previous related works.

Multitasking network architectures can be broadly classified into two. Hard parameter sharing networks and soft parameter sharing networks.



(a) Soft parameter sharing. (b) Hard parameter sharing.

Fig. 1. Multi tasking architectures.

Soft parameter architecture contains identical independent models performing different tasks in parallel, where their weights are updated with a few regularisation constraints as in Fig 1a, while the hard parameter architecture uses a set of layers shared between different tasks as in Fig 1b. From a performance perspective, hard parameter sharing models are better, as they have shared layers, reducing the number of trainable parameters and computation requirements. Hence we chose the hard parameter architecture for the proposed model as it is more suitable for real-time systems and edge devices.

The multitasking architectures have been explored by researchers in computer vision, natural language processing, reinforcement learning domains. [2] used an encoder-multi-decoder architecture for depth estimation as well as both instance and semantic segmentation. They also suggested giving weightage to losses calculated for different tasks by determining the uncertainties in the output of each task. [3] proposed to use head pose estimation and other facial attributes as auxiliary tasks to improve detection of facial landmarks. In order to prevent overfitting of the model to any of the auxiliary tasks, the authors also suggested using task wise early stopping. FT-MTL-NET (Feature Transfer – Multi Task Learning – Net) [4], a model developed for medical imaging applications, suggested using feature transfer learning between branches of the multitask model. [5] used YOLO (You Only Look Once) model to detect the cancerous masses and then a full resolution convolution network (FrCN) to segment the exact area of occurrence as well as classify it as benign or malignant. Recently, mask SSD (Single Shot Detector) was proposed [6] to enhance the performance of the popular SSD model to detect small objects. The model contains a detection branch and a segmentation branch to enhance the extracted features with contextual information.

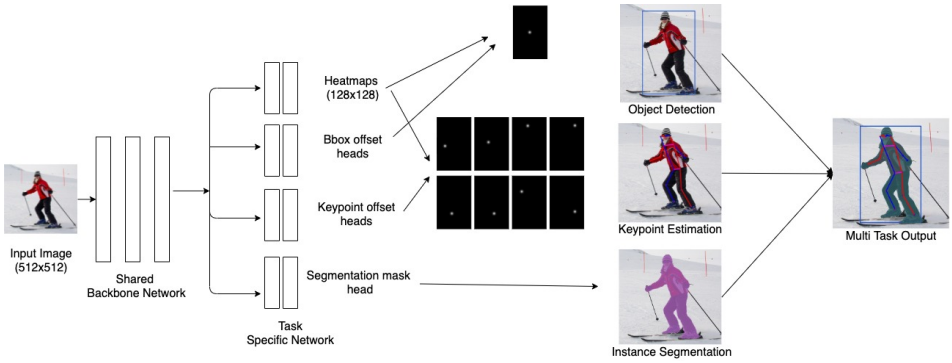


Fig. 2. Multi task architecture.

Multitask learning has been applied more in Natural Language Processing (NLP) than computer vision. [7] used multitask learning and suggested that auxiliary tasks like parts of speech tagging and entity recognition require low-level supervision. A hierarchy for multiple NLP tasks was predefined in [8]. [9] used a multitasking approach for search and retrieval of websites. They used representation learning using a multitask network for information retrieval related to the web search and identifying web query semantic classification. Taskonomy [10] developed a visual space for over 20 different tasks by computing the transfer learning dependencies both in 2D and 3D space. Natural Language Decathlon [11] is a multitasking challenge to perform ten different NLP tasks such as question-answering, machine translation, summarization, sentiment analysis *etc*. [12] used multitasking models in medical fields for drug discovery. A higher predictive accuracy was obtained for the multitasking model than the single task models. Empirical studies showed that the overall performance of multitask networks improved as additional tasks and data were added.

3. PROPOSED MULTI TASKING MODEL

This work proposes a novel model based on multi-tasking architecture that performs object detection, instance segmentation and keypoint estimation tasks. We used a single-stage anchorless model Centernet [13] as our baseline. It considers the objects as a single keypoint (the center of the bounding box). Once the centre point is detected, Centernet regresses to all other object properties, such as size, 3D location, orientation *etc*.. Different variants of Centernet with different backbone networks such as resnet, Deep Layer Aggregation (DLA), Hourglass are available with varying accuracy and computation requirements. Prior to Centernet, models like CornerNet [14] and ExtremeNet [15] also had similar approaches. CornerNet performs object detection by identifying the two opposite corner points of the bounding box. Instead of two opposite points, Extremenet identifies all four corner points through the keypoint estimation method.

Fig. 2 shows a high-level architecture of the proposed model. We modified the Centernet baseline model by replacing the backbone network (shared layers) with a Harmonic Densely Connected Network (HarDNet) [16]. HarDNet is a simple U shaped network

with Conv3x3, ReLU, bilinear interpolation upsampling, and Sum-to-1 layer normalization. We chose HarDNet as the shared feature extraction layer as we want our model to work on both high-end GPUs and edge devices like Jetson boards. The model has optimised DRAM memory traffic by using Nvidia profiler and ARM Scale-Sim software. The two variants, HarDNet 68 and HarDNet 85, achieved 76.5 and 78.0 Top1 accuracy on the Imagenet classification dataset.

The features extracted by the shared network are given to the task-specific layers. We modified the baseline model to predict 17 body keypoints from the COCO dataset and the object's centre. The COCO keypoints include the nose, left eye, right eye, left ear, right ear, left shoulder, right shoulder, left elbow, right elbow, left wrist, right wrist, left hip, right hip, left knee, right knee, left ankle, right ankle. The keypoints are predicted as heatmaps of size 128x128 for an input image size of 512x512 (with a stride size of 4). The model also predicts an offset error in both x and y axes to compensate for the uncertainty of the keypoint while upscaling from 128x128 to 512x512 input size. As in the original Centernet model, our model has two regressive heads to predict the height and width of the model. We added another task head based on conditional convolution networks for instance segmentation as an auxiliary task. The CondInst [17] uses dynamically generated instance aware segmentation mask heads. Unlike other segmentation models, they use a compact fully convolutional layers to generate the masks without relying on expensive ROI operations. Variants of CondInst claimed 35.7 mAP to 40.2 mAP on COCO segmentation tasks in the published paper. The segmentation mask head in our architecture consists of three 3x3 convolutions with relu activation except for the final layer. We chose eight channels for the mask head as suggested in the study [17].

3.1 Multi Task Loss

Defining the loss function of the multitasking network is the core idea of joint learning/multitask learning. Individual loss is calculated for each task-specific head and can be used to optimise the model separately or jointly. Previous approaches used alternate training on the same model with different tasks. After calculating the losses, the optimisers are called for each task one after the other. Our approach is, however, a more simpler one. We computed the sum of the loss of individual tasks for each iteration and optimised the model with this joint loss. Combining different losses will be beneficial in these ways:

- Acts as regularization, smoothing the loss function.
- Gives informative priors for the model: Prior knowledge on a related task can significantly improve the learning curve.
- Robust to data specific noises: Models find a more generalised representation of the data.
- Self Feature Attention: If a feature is relevant to more than one task, it should be a more important and better representative of the data. The model will be able to prioritise them over other features.
- Provides better and faster convergence: Similar to transfer learning helping the models learn better in another related domain, Jointly learning different tasks helps the model learn better and faster.

- Avoids over-fitting: Losses of additional tasks can help avoid over-fitting on individual tasks or data.

Our multi-task loss is a combination of pixelwise logistic regression with focal loss [13] (heatmap heads), regression loss (bbox height, bbox width and offset heads) and dice loss [18] (segmentation head) as in Eq. (1).

$$L_{Tot} = L_{heat} + \lambda_{reg} * L_{box} + \lambda_{segm} * L_{dice} \tag{1}$$

where L_{box} is regression loss for bounding box height and width.

$$L_{heat} = \Sigma_N(\lambda_{heatmap} * L_{focal} + \lambda_{offset} * L_{offset}) \tag{2}$$

$$L_{focal} = \frac{-1}{N}\Sigma \begin{cases} (1-\hat{Y})^{\alpha}\log\hat{Y} & \text{if } Y = 1 \\ (1-Y)^{\beta}(\hat{Y})^{\alpha}\log(1-\hat{Y}) & otherwise \end{cases} \tag{3}$$

We took α =2 and β=4 as seen in Cornernet [14].

$$L_{dice} = \frac{2 * Number\,of\,overlapping\,pixels}{Total\,number\,of\,pixels} \tag{4}$$

While training with the combined loss of different tasks, a few considerations must be made.

- There is a chance for the gradients from different tasks to interfere negatively, causing the loss to fluctuate and making the learning process unstable.
- There can be cases where one task dominates the others, causing the final loss to be lower but with some underperforming tasks.

We overcome these issues by weighing the different task losses before combining them. Empirical analysis gave the value of $\lambda_{heatmap}$, λ_{reg}, λ_{offset} and λ_{segm} as 1, 0.05, 0.5 and 0.1 respectively.

4. TRAINING AND EVALUATION

We trained two variants of the proposed model, the h85 model with a hardnet85 backbone and a lighter h68 model using a Hardnet68 backbone. The training methodology is extensively detailed in [1]. During inference, We use a 3x3 max pooling on the predicted heatmaps to find the peaks that denote the center of the object. It is highly unlikely for the center to be at the same point, even for multiple objects of the same class. The intensity of the peak of heatmaps gives confidence for the detection and keypoints. Thus we can avoid costly post-processing like Non-maximum suppression (NMS) [19] algorithm, traditionally used to prevent duplicate boxes for an object.

Tables 1 shows the evaluation results of the proposed multitasking model on the three different tasks it is trained for. The Average Precision (AP) is a popular metric in

Table 1. COCO evaluation results.

Task	Model Name	AP	AP 50	AP 75	AP small	AP medium	AP large
Object Detection	Ours-h68	0.368	0.582	0.397	0.164	0.556	0.498
Object Detection	Ours-h85	0.412	0.599	0.452	0.131	0.503	0.709
Segmentation	Ours-h68	0.295	0.543	0.297	0.102	0.452	0.437
Segmentation	Ours-h85	0.382	0.643	0.406	0.135	0.524	0.638
Keypoint Estimation	Ours-h68	0.393	0.643	0.408	-	0.428	0.418
Keypoint Estimation	Ours-h85	0.530	0.786	0.563	-	0.449	0.664

Table 2. Object detection task – Comparison with baseline.

Model Name	AP	AP 50	AP 75
Resnet50+CondInst	0.358	0.540	0.384
HardNet68+SSD512	0.317	–	–
HardNet85+SSD512	0.351	–	–
Centernet-Hourglass	0.403	0.591	0.440
Centernet-DLA1x	0.363	–	–
Centernet-DLA2x	0.374	0.551	0.408
Centernet-Resnet101(DCN)	0.346	0.530	0.369
Centernet-Resnet18(DCN)	0.281	0.449	0.296
Extremenet+Hourglass	0.358	–	–
Extremenet+DeepLayerAggregation (DLA)	0.330	–	–
Ours-h68	0.368	0.582	0.397
Ours-h85	**0.412**	**0.599**	**0.452**

object detection. Intersection over Union (IoU) is the ratio of the area of overlap and area of union between two bounding boxes. The precision and recall are calculated with different IoU thresholds. We calculated the AP over different IOU thresholds, AP at IOU thresholds 0.5(AP50) and 0.75 (AP75). The HarDNet85 (h85) backbone network model gave 43.6 AP on object detection, 38.2 AP on instance segmentation, and 53.0 AP on keypoint estimation tasks on evaluation with coco validation data. The smaller model with HarDNet68 (h68) backbone network gave 36.8 AP on object detection, 29.5 AP on instance segmentation and 39.3 AP on keypoint estimation tasks. These results are evaluated without Flip/Multiscale testing.

4.1 Comparison with State of the Arts

Our proposed model predicts object detection, keypoint estimation and instance segmentation in a single forward pass. As far as we know, there haven't been any previous works in which a single model predicts these exact three computer vision problems together. Thus a direct comparison between the models is not possible. Tables 4-5 compare the performance of different tasks individually with their baseline model variants trained for a single task.

In Table 2, we can see that the bounding box AP for our h85 multi-tasking model is better than the original conditional instance segmentation model with resnet, SSD model with similar hardnet backbone networks, variants of Centernet and Extremenet models.

Table 3. Segmentation task – Comparison with baseline.

Model Name	AP	AP 50	AP 75	AP small	AP medium	AP large
Resnet50+CondInst	0.354	0.564	0.376	**0.184**	0.379	0.469
Ours-h68	0.295	0.543	0.297	0.102	0.452	0.437
Ours-h85	**0.382**	**0.643**	**0.406**	0.135	**0.524**	**0.638**

Table 4. Keypoint estimation task – Comparison with baseline.

Model Name	AP	AP 50	AP 75	Speed (Fps)
Hourglass-104	**0.640**	–	–	6.6
DLA-34	0.589	–	–	23
Ours-h68	0.393	0.643	0.408	**41.66**
Ours-h85	0.530	0.786	0.563	**32.26**

The lighter multi-tasking model has comparable performance with the best-performing Centernet baseline models. Table 3 compares our models with the baseline conditional instance segmentation model for the instance segmentation task. The h85 model outperformed the baseline model on AP at different IOUs and for medium and large objects. Table 4 shows that we achieved acceptable performance in keypoint estimation task with less than half the computation time required for other models. The non optimised version of our model performs more than five times faster than the current state of the art Hourglass model in estimating the keypoints.

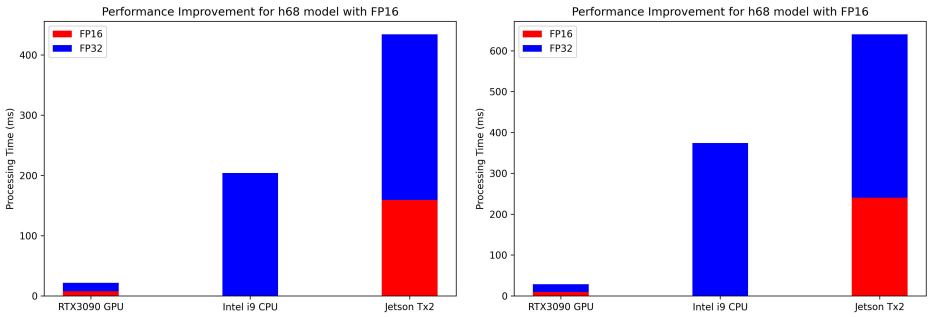
5. OPTIMISATIONS AND PERFORMANCE EVALUATION

We used different configurations to assess the model's performance, including 10th gen Intel Core i9-10900K CPU, RTX3090 GPU, and Nvidia Jetson TX2 board. We optimised the model by fusing the convolution layers and batch normalisation layers. As both convolution and batchnorm are linear operations to the data point x, and they can be rewritten in terms of matrix multiplications: $T_{bn} * S_{bn} * W_{Conv} * (x)$, where we first convolve (W) the data (x), then scale (S) and time shift (T) it using the batchnorm-trained parameters.

Finally, We optimised the models using TensorRT 8.2.2.1 and experimented with FP32 and FP16 quantization. The performance results are given in Table 5. The model got nearly 2x improvement when converted to float16 weights. A visualisation of the performance results on different system configurations is shown in Figs. 3 and 4.

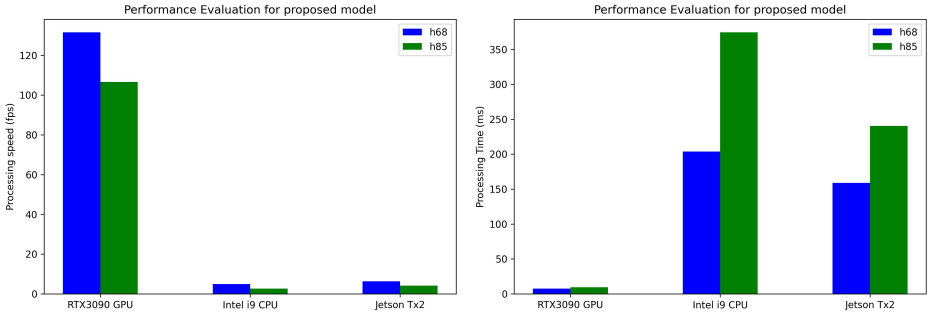
Table 5. Performance evaluation of the proposed model.

Model Name	RTX3090 GPU (FP32)	RTX3090 GPU (FP16)	Intel i9 CPU (FP32)	Jetson Tx2 (FP32)	Jetson Tx2 (FP16)
Ours-h68	014.11 ms	007.60 ms	203.82 ms	275.28 ms	158.97 ms
Ours-h85	018.88 ms	009.38 ms	374.36 ms	399.89 ms	240.34 ms



(a) Improvement for h68 model. (b) Improvement for h85 model.

Fig. 3. Performance improvement with FP16 quantization.



(a) Processing speed. (b) Processing time.

Fig. 4. Performance evaluation on different devices.

On analysing the performance on various devices, the proposed model has sufficient processing speed and performs multiple computer vision tasks simultaneously. As future work, We look forward to optimise the models further by using int8 quantisation and calibration on more resource-constrained edge devices.

6. CONCLUSION

Most of the existing state of the art models are optimised for certain benchmark parameters on curated datasets for specific tasks. Actual industrial use cases often require results from multiple models, used sequentially or as an ensemble, for generating meaningful business insights. The final output can only be obtained after getting the results from the slowest model among them. Due to uneven resource requirements for the different models, the allocated resources will mostly be underutilised. Also, each model extracts its own features and cannot take advantage of features pre-computed by other models. This, in addition to the pre and post-processing overheads associated with each model, is sub-optimal for a real-time video analytics pipeline. In most cases, the pre-post processing overheads are done on CPUs, causing the processing pipeline to throttle.

We explored the multitasking architecture for neural networks to overcome these issues. We developed a multitasking model that can do multiple tasks in a single forward pass. The joint learned model can do multiple related tasks – object detection, keypoint estimation for body posture analysis and instance segmentation for improved tracking.

We compared our model with the state of the art single task models that had similar architectures to our baseline, as we couldn't find models developed for exactly the three tasks that we trained for. We see that our models outperformed in terms of accuracy in the bounding box (41.2 mAP) and segmentation tasks (38.2 mAP) on COCO dataset. We achieved comparable performance in the keypoint estimation task (53.0 mAP) with nearly 5-7 times faster inference time than others. We attribute the improved performance of the model to the carefully chosen multitasking architecture and the optimised backbone network. The use of auxiliary tasks enabled the model to become a more robust and generalised one, by understanding the complex relationship between the tasks and reducing the chances of overfitting. As future work, we are planning to implement the multitasking model in previously developed robotic testbed [20, 21] to measure the impact of the additional auxiliary tasks through an ablation study.

We optimised the model by combining batch norm layers with convolution layers and applying float16 quantisation for the weights. We achieved 107 frames per second (fps), while a lighter version achieved 131fps on RTX 3090 GPU. We also benchmarked the models on Nvidia Jetson Tx2 and got 4.2 fps and 6.3 fps for the two respective models. We plan to use int8 quantisation and calibration to further improve the processing speed on edge devices. Using a multitasking model helped us avoid using multiple models for different use cases and reduce their pre-post computational cost and time complexity. This enabled us to optimise the video analytics pipeline and process data in real time with sufficient accuracy levels.

REFERENCES

1. A. J. Chemmanam and B. A. Jose, "Joint learning for multitasking models," *Responsible Data Science*, Springer, 2022, pp. 155-167.
2. A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7482-7491.
3. Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *Proceedings of European Conference on Computer Vision*, 2014, pp. 94-108.
4. F. Gao, H. Yoon, T. Wu, and X. Chu, "A feature transfer enabled multi-task deep learning model on medical imaging," *Expert Systems with Applications*, Vol. 143, 2020, p. 112957.
5. M. A. Al-Antari, M. A. Al-Masni, M.-T. Choi, S.-M. Han, and T.-S. Kim, "A fully integrated computer-aided diagnosis system for digital x-ray mammograms via deep learning detection, segmentation, and classification," *International Journal of Medical Informatics*, Vol. 117, 2018, pp. 44-54.
6. C. Sun, Y. Ai, S. Wang, and W. Zhang, "Mask-guided ssd for small-object detection," *Applied Intelligence*, Vol. 51, 2021, pp. 3311-3322.

7. A. Søgaard and Y. Goldberg, "Deep multi-task learning with low level tasks supervised at lower layers," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, Vol. 2, 2016, pp. 231-235.
8. K. Hashimoto, C. Xiong, Y. Tsuruoka, and R. Socher, "A joint many-task model: Growing a neural network for multiple nlp tasks," *arXiv Preprint*, 2016, arXiv:1611.01587.
9. X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang, "Representation learning using multi-task deep neural networks for semantic classification and information retrieval," in *Proceedings of the 53th Annual Meeting of the Association for Computational Linguistics*, 2015, pp. 912-921.
10. A. R. Zamir, A. Sax, W. B. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3712-3722.
11. B. McCann, N. S. Keskar, C. Xiong, and R. Socher, "The natural language decathlon: Multitask learning as question answering," *arXiv Preprint*, 2018, arXiv:1806.08730.
12. B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, "Massively multitask networks for drug discovery," *arXiv Preprint*, 2015, arXiv:1502.02072.
13. X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv Preprint*, 2019, arXiv:1904.07850.
14. H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of European Conference on Computer Vision*, 2018, pp. 734-750.
15. X. Zhou, J. Zhuo, and P. Krahenbuhl, "Bottom-up object detection by grouping extreme and center points," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 850-859.
16. P. Chao, C.-Y. Kao, Y.-S. Ruan, C.-H. Huang, and Y.-L. Lin, "Hardnet: A low memory traffic network," in *Proceedings of IEEE/CVF International Conference on Computer Vision*, 2019, pp. 3552-3561.
17. Z. Tian, C. Shen, and H. Chen, "Conditional convolutions for instance segmentation," in *Proceedings of the 16th European Conference on Computer Vision*, Part I, 2020, pp. 282-298.
18. R. Deng, C. Shen, S. Liu, H. Wang, and X. Liu, "Learning to predict crisp boundaries," in *Proceedings of European Conference on Computer Vision*, 2018, pp. 562-578.
19. A. Neubeck and L. Van Gool, "Efficient non-maximum suppression," in *Proceedings of IEEE 18th International Conference on Pattern Recognition*, Vol. 3, 2006, pp. 850-855.
20. P. Nithin, A. J. Chemmanam, B. A. Jose, J. Mathew *et al.*, "Interactive robotic testbed for performance assessment of machine learning based computer vision techniques," *Journal of Information Science and Engineering*, Vol. 36, 2020, pp. 1055-1067.
21. P. Nithin, A. Francis, A. J. Chemmanam, B. A. Jose, and J. Mathew, "Face tracking robot testbed for performance assessment of machine learning techniques," in *Proceedings of IEEE 7th International Conference on Smart Computing and Communications*, 2019, pp. 1-5.

Ajai John Chemmanam currently a Research Scholar in Artificial Intelligence and Machine Learning at the Department of Electronics, CUSAT, Kerala, India. He obtained his Masters and Bachelors degree from Cochin University of Science and Technology, Kerala, India. He has three years of industrial experience in AI and ML. He is also an active contributor to various open-source projects. For his research contributions, he was awarded the Prime Minister's Fellowship for Doctoral Research in 2020.

Bijoy A Jose currently working as an Associate Professor in the Department of Computer Science, CUSAT, Kerala, India. He has received his B.Tech from the School of Engineering CUSAT and MS from the State University of New York. During his graduate studies, he has been an intern with IBM, New York and Intel Corporation, Illinois. He received his Ph.D. from Virginia Tech and worked in Intel Corporation in California and Bangalore for four years. He received the Early Career Research Award from the Department of Science and Technology, Govt. of India, in 2016. He is the Principal Investigator for multiple funded projects from DST and IEEE. His areas of interest include cyber security, the Internet of Things and cyber-physical systems.

Asif Moopan is the Founder and CEO of Vuelogix, an AI and IoT solutions company based out of Bangalore. He drives the product and technology roadmap for the company. Prior to Vuelogix, he was the co-founder and CTO of Carinov, which was into Digital Media Narrowcasting and Audience Measurement. He has over sixteen years of experience in technology and started his career as an embedded system and DSP engineer at Motorola and Tata Elxsi.