Deep Learning in Robotic Vision

Jack Pay

March 25, 2024

1 Literature Review

Development in Deep Learning (DL) has spurred significant progress in various robotics fields, in particular robot vision [20]. DNNs can easily model complex functions without requiring expert knowledge or feature engineering [20]. However, numerous challenges persist within this field, such as models' adaptability to spatially and temporally changing environments, something key to robotic vision as opposed with computer vision [25, 1]. This review will aim at analysing the state of the art DL methods for robotic vision, summarising the explosive advancement made in recent years.

Classified by their Convolutional layers, Pooling layers, Fully-Connected layers and non-linear activation functions [3], Convolutional Neural Networks (CNNs) are a staple of robot vision [22]. Architectures such as AlexNet [11], ResNet [7] and EfficientNet [27] have revolutionised robotic vision, allowing significant advancement in performance throughout the past few decades.

In 2012, Krizhevsky et al. developed AlexNet [11], a CNN architecture that achieved state of the art performance in image classification at that time and won the ImageNet LSVRC-2010 competition [11]. AlexNet was made distinct by its use of dropout, a regularization method used to randomly omit hidden units from the training process [8]. Counter-intuitively, this prevents overfitting [24] and produces models that generalize better [12]. This is because it forces hidden units to not always rely on the prescence of other specific units [8]. AlexNet also utilised a convolution implementation specifically produced to take advantage of GPUs and accelerate training [11].

Winning the ILSVRC-2015 classification task, ResNet was developed in 2015 by He et al. [7]. Similarly to AlexNet, this CNN for image classification was trained and evaluated upon the ImageNet dataset but owed its success to the "Residual connections" employed throughout its architecture. Residual connections are a type of shorcut, or skip connection, where the output of a hidden layer is combined with the original input to that layer [7]. This makes DNN significantly easier to train as it reduces the vanishing gradient problem [2] and therefore improves training and test loss [13].

However, a primary downside of CNN architectures for robotics at this time was the size of models produced [5]. The size of models and, therefore, speed of inference is a crucial factor within robot vision [6], where a difference of seconds could result in or prevent human injury.

Developed in 2019 by Tan et al., EfficientNet [27] was a CNN that addressed this issue. EfficientNet employed a method called "compound coefficient" for scaling the size of the DNN (in terms of width, depth and resolution) based on the size of the input [27]. Compound coefficient was massively successful for developing an architecture not only with state of the art performance but also computational efficiency and improved model size, being able to run within even smaller, less powerful devices [1, 17]. EfficientNet also boasted further, faster training times [26], a quality useful for incremental learning within robotics [25].

Predominantly employed within the field of NLP [18, 9], transformer architectures have also shown great success in the field of robotic vision [21]. Examples of architectures include the original Vision Transformer (ViT) [4], the Visual Saliency Transformer [16] and Contextual Transformer [14].

Transformer architectures boast several advantages when compared to other methods: from their state of the art performance yet smaller computational cost [4], to requiring fewer prior assumptions or inductive biases prior to training [9]. Fine-tuning pretrained models, rather than training one from scratch, reduces the amount of data and effort required for training transformers [23]. A major part of robotics, vision or otherwise, is incorporating and utilising data of different modalities [29]. Transformers are able to easily extend and incorporate different modalities [9] such as combining both visual and proprioceptive senses for improved locomotion [28].

In summary, DL is the state of the art method for robotic vision. CNNs and transformers are intrinsic to robotic vision, addressing robotic's crucial requirements for accuracy, speed and efficiency [25, 6].

2 Introduction

Note that all experimentation was conducted within the notebook found here: `https://colab.research.google.com/drive/1SfKWE0-IMnT1rEtc4YB5jPUos6SVVEtE?usp=sharing`.

This experimentation aimed at finding a high performance DNN architecture for image classification, therefore researching robotic vision. For this aim, the CIFAR100 [10] dataset was employed due to its easy accessibility, free usage, complexity and size. This dataset poses a higher level of complexity compared with CIFAR10 [10], whilst CIFAR100 involves classification of 100 classes, rather than just 10. All images within the CIFAR datasets are 32x32 pixels in size and have 3 colour channels. Not that other than normalisation, little preprocessing was carried out on this dataset.
In total, sixteen experiments were conducted experimenting with the architecture of CNNs, their structure being the primary experimental hyperparameter. Experiments varied the components used within the model architecture, trialing the added benefit of Residual connections and Multi-Head Attention (MHA). Furthermore, investigation into the network's width and depth was carried out. The methodology involved utilising KerasTuner [19] to carry out a quadratic grid based search upon the number of units within each Convolutional layer and the number of Residual blocks in the network, simultaneously.

3 Approach

Presented within Table 1 is a summary of the major experiments conducted within this research.
Each of the experiments utilised a Learning Rate (LR) of 0.0001 (allowing slow convergence towards the optimal) and trained for 100 epochs, allowing for models to reach an optimum. The Root Mean Square Propagation (RMSprop) optimization algorithm was also employed to tune the LR throughout training and to circumvent under and overfitting. CIFAR100 consists of 10,000 test data samples, set aside for final evaluation of each trained model, and 50,000 training samples. These training samples were then split: 80% being set aside for training and the remaining 20% for validation. The validation set was utilised to evaluate the performance of and tune the hyperparameters of the networks throughout training.
Firstly a basic CNN architecture consisting of Convolutional layers, Max-Pooling layers and a Fully-Connected final layer was trained to serve as a baseline for the experimentation. Next, for the second experiment, this architecture was adapted: after a first plain CNN block, Residual blocks were used. These consisted of Convolutional + Activation + Batch Normalisation + Convolutional + Add + Activation + Batch Normalisation + Max Pooling, where the Add layer summed the original input with the hidden representation from the previous layer. Experiment three integrated a MHA block into the base CNN architecture. The MHA blocks consisted of MHA + Add + Layer Normalisation, where the Add layer summed the original input and output from the Attention layer. Next, the fourth experiment utilised a combination of the architectures from the previous two experiments: both the Residual and MHA blocks were added into a CNN, further boosting the performance for image classification.
The fifth experiment used KerasTuner's GridSearch [19] to run twelve individual experiments, varying the width of the Convolutional layers and the number of Residual blocks. The values [32, 64, 128, 256] and [1,2,3] were employed respectively, changing the width and depth of the network. These experiments involved training the models for just twenty epochs and with a LR of 0.0001 to find the optimal model architecture. This number of epochs was a compromise between saving resources and ensuring reliability of the experiment. The result of these twelve experiments revealed the best width to be 256 Convolutional units and the best depth to be 3 Residual blocks.
Once the fifth experimentation had concluded, the best hyperparameters were employed to train a final model. This model was further optimised using Exponential LR scheduling, some minor Data Augmentation and Early Stopping. The effect of this was further boosted performance and reduced overfitting.

Experiment	Summary	Testing Accuracy	Testing Loss
1	Basic CNN	34.960%	2.560
2	CNN with Residual connections	47.480%	2.056
3	CNN with MHA	46.900%	2.123
4	CNN with Residual connections and MHA	49.030%	1.969
5	CNN with Residual connections and MHA. Varied size of the Convolutional layers with the values [32, 64, 128, 256] and varied the number of Residual blocks with the values [1,2,3]	(Best) 57.410%	(Best) 3.394
6	CNN with Residual connections and MHA. Exponential LR, Data Augmentation and Early Stopping were used	61.130%	1.977

Table 1: A description of each experiment presented with their test accuracy and loss. Note that for experiment 5, twelve experiments were conducted. Documented above are the metrics for the best architecture found.

4 Evaluation

This section will carry out further, detailed evaluation of the results from each of the experiments.
Shown in Figure 2, are the primary experiments. Experiments 1–4 involved modifying the CNN architecture. Both adding the Residual and MHA blocks improved performance of the CNN architecture, the former providing a larger increase than the latter. These conclusions are well-trodden: the benefit of skip-connections being evident through ResNet [7, 2] and Attention through Transformers [18, 4]. The Residual connections allow further information to be propagated throughout the network, reducing the vanishing gradient problem and increasing performance. The addition of Attention allows the model to select more important aspects of the image, mirroring animal vision.
Potentially more interesting is experiment 5 and the results of the twelve experiments upon the size of the model architecture, shown in Figure 3. Interestingly, the size of the architecture makes a huge impact on the performance of the architectures. A smaller width of network, such as using 32 units within each Convolutional layer results in far lower performance and slower convergence towards an optimal model. This width results in a model that plateaus in performance around 48% accuracy. But using a wider network, such as up to 256 units per layer, creates a model with improved performance and faster convergence, despite the increased number of network parameters. However, models of this size present a double-edged sword. The longer these larger models are trained, the more they can tend to overfit. This can even result in a decrease in overall performance: the neural network becoming so focused on the training data that the validation loss greatly worsens. This is evident in Figure 2: after approximately 20 epochs, experiment 5 increase in loss again, becoming one of the most accurate models but also the one with the highest loss.
The primary reason for this is the time taken for convergence and the LR scheduling. The smaller networks took a long time to converge, barely reaching a plateau in performance by the end of the hundredth epoch. However, the larger models quickly increase in performance, possibly faster than the LR scheduler can keep up. The LR is still too high at this point, resulting in overfitting.
From experiment 5, the optimal hyperparameters, of 256 Convolutional units and 3 Residual blocks, were discovered. This provided the lowest loss of any of the twelve experiments. From these findings it was also discovered that the width of the neural network made a far more significant impact compared with the depth. Varying the number of Residual blocks in a row would have different impacts of the performance: sometimes increasing accuracy and other times decreasing it. Overall, the number of units in each convolutional layer made a far larger and consistent impact. This is useful for finding a smaller neural network for robotic vision. It may be better to prioritse the width rather than depth or even find a suitable balance between the two whilst speed and size of such models is a crucial in robotics.

4.1 Optimal Hyperparameters

With the findings from the previous experiments, a final model was then trained. The architecture for this model is presented within Figure 4. This model utilised the best hyperparameters of 256 Convolutional units and 3 Residual blocks. Both the Residual blocks and MHA block were incorporated into this architecture.
Furthermore, with the findings from experiment 5, surrounding overfitting of such a large model, a second LR scheduling method of Exponential Decay was utilised. Whilst RMSprop algorithm was previously used to adapt the LR throughout training, this was found to not be enough. Therefore Exponential Decay was also employed to decrease the overall LR at regular intervals. With this method, every epoch the LR was multiplied by $e^{-0.99}$, to ensure further stability within the training process. To balance the fast momentum of this scheduling, a higher initial LR of 0.001 was employed.
This resulted in a final model that plateaued with a higher test accuracy and lower loss of 61.130% and 1.977 . Rather than overfitting after 20 epochs, the model continued to learn the structure of the data.
Whilst the performance of the final model is good for

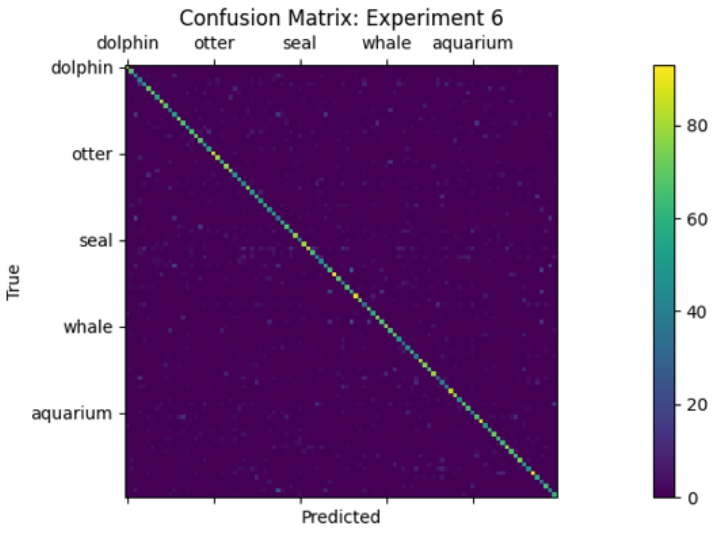

Figure 1: A confusion matrix of the final model's performance on the CIFAR100 test dataset. This CNN was composed of Connvolutional layers with 256 units, 3 Residual blocks followed by a MHA block with 3 heads. This architecture achieved an accuracy of 61.130% and loss of 1.977 .

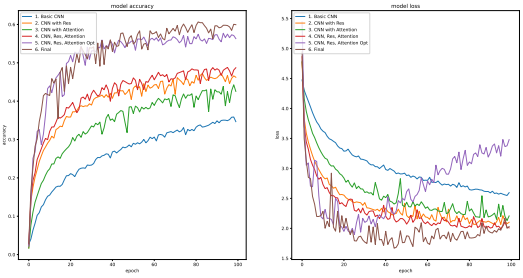

Figure 2: The different major architectures. Experiment 1 used a basic CNN, 2 a CNN with Residual connections and 3 a CNN with MHA. Experiment 4 consisted of a CNN with 32 wide Convolutional layers, 2 Residual blocks and a MHA block. Experiment 5 and 6 here consisted of a CNN with 256 wide Convolutional layers, 3 Residual blocks and a MHA block. Experiment 6 was a further optimised model, utilising methods such as Exponential LR scheduling to reduce overfitting.

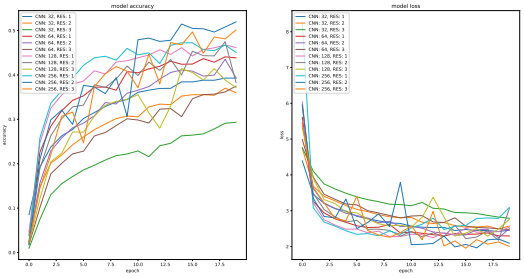

Figure 3: The effect of different widths and depths on the validation accuracy and loss of the model architecture. Here the size of the Convolutional layers and the number of Residual blocks were varied using the values [32, 64, 128, 256] and [1,2,3] respectively.

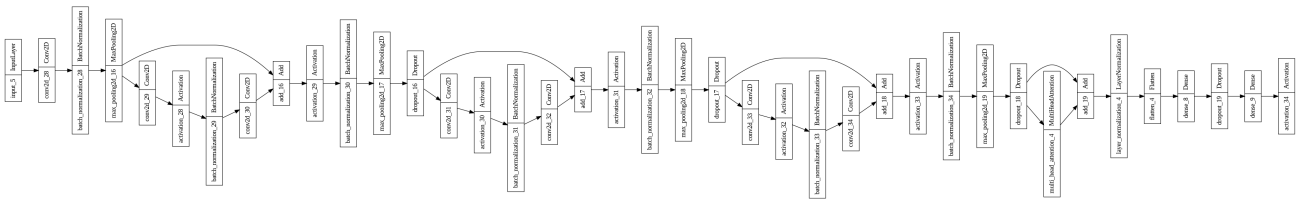

Figure 4: The architecture of the final model. This is a CNN with Residual connections and Multi-Head Attention (MHA). Convolutional layers contain 256 units and the architecture contains 3 main Residual blocks.

such a complex and high dimensional classification problem, it could still further be improved. Therefore, an extensive confusion matrix was generated, presented within Figure 1. This shows a summary of the performance of the final model on the CIFAR100 dataset. This was produced to further analyse potential difficulties the model faced, with the aim of suggesting possible future improvements.

Some of the CIFAR100 classes achieved a high F1 score, such as the Beaver class with 0.832 or Apples class with 0.875. However, other classes achieved quite a low performance, possible therefore being responsible for the overall lower performance of this model. Examples include the Girl class with an F1 score of 0.291 or the Camel class with an F1 score of 0.312.

A primary reason for this low score could be related to the data distribution and image quality for these classes. For example, other than the Girl class there are several alternative human related classes within CIFAR100 , such as Woman, Man and Boy. Both Woman and Man share a similarly low F1 score of 0.440. It may be that the data samples belonging to each of these classes are quite similar, making it harder for the model to draw a valid separation between what is a Girl and what is a Woman. Context clues such as wrinkles, face size or clothes may be difficult to make out in the low resolution images. The Camel class' low metrics could also further support this claim, whilst Camel samples might be quite similar to Cattle or even Lion samples. These are all quadrupedal, tan coloured animals, after all.

4.2 Future Work

Firstly, this research could be taken further, experimenting with wider and deeper CNN architectures. Here the limit of 256 units was used for the Convolutional layers, however this size could be further expanded. 512 or 1024 units could be trialled to assess their performance also. This research was limited by time, cost and computation power making it infeasible to trial architectures of such sizes.

Furthermore, different combinations of widths could be trialed. Here each Residual block was the same width, having Convolutional layers with the same number of units. Instead, different combinations could be tried to see, for example, if a network with a narrow front and wide end is beneficial. This would exponentially add complexity, requiring an even longer search algorithm.

This research found that larger networks performed better, but this is maybe harmful to robotic vision. As mentioned within the earlier sections, the size and speed of robotic vision models is crucial, with slow models

potentially causing financial damage or harm to humans. Further work could explore the speed of the models presented here, looking to better balance the efficiency of models and the time required for inference.
Another improvement of this work could be to better visualise how the model distributes and separates classes, employing methods such as T-distributed Stochastic Neighbor Embedding. Currently, analysis of the final model is mostly speculative and therefore could be further enhanced with sufficient proof of how the model represents samples within the high dimensional, semantic space. This could help to focus on which classes potentially require more data or which features are crucial for separating the data samples.
Most investigation here focused on Residual connections and making a CNN of different sizes. Future work could also expand on the other elements introduced within this work, aiming to exploit them and improve performance overall. Multi-Head Attention (MHA) was employed in experiment 3, increasing performance of image classification but otherwise, this method was not touched upon greatly, due to time constraints and an already quite complex optimisation task. Different numbers of MHA heads could be trialed or Attention at different points throughout the network could be attempted. Selective attention is a major part of human and robotic vision, therefore further focus on this could be beneficial.
Finally, whilst it poses many benefits, the CIFAR100 dataset is also not perfect. This dataset contains mostly quite simple images, with a single target in frame and clear, complete images. The real world is far noisier than this, with multiple potential target in view at the same time or peculiar poses. Robot vision must be able to cope with these different perspectives and understand how to better orient themselves, or an object being viewed, to maximise performance. Therefore a different, harder dataset such as COCO [15] could be trialed instead.

References

[1] Mouna Afif et al. "An evaluation of EfficientDet for object detection used for indoor robots assistance navigation". In: *Journal of Real-Time Image Processing* 19.3 (2022), pp. 651–661.

[2] Lokesh Borawar and Ravinder Kaur. "ResNet: Solving Vanishing Gradient in Deep Networks". In: *Proceedings of International Conference on Recent Trends in Computing: ICRTC 2022*. Springer. 2023, pp. 235–247.

[3] Anamika Dhillon and Gyanendra K Verma. "Convolutional neural network: a review of models, methodologies and applications to object detection". In: *Progress in Artificial Intelligence* 9.2 (2020), pp. 85–112.

[4] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[5] Azeddine Elhassouny and Florentin Smarandache. "Trends in deep convolutional neural Networks architectures: a review". In: *2019 International Conference of Computer Science and Renewable Energies (ICCSRE)*. 2019, pp. 1–8. DOI: 10.1109/ICCSRE.2019.8807741.

[6] Manabu Hashimoto, Yukiyasu Domae, and Shun'ichi Kaneko. "Current status and future trends on robot vision technology". In: *Journal of Robotics and Mechatronics* 29.2 (2017), pp. 275–286.

[7] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[8] Geoffrey E. Hinton et al. "Improving neural networks by preventing co-adaptation of feature detectors". In: *CoRR* abs/1207.0580 (2012). arXiv: 1207.0580. URL: http://arxiv.org/abs/1207.0580.

[9] Salman Khan et al. "Transformers in vision: A survey". In: *ACM computing surveys (CSUR)* 54.10s (2022), pp. 1–41.

[10] Alex Krizhevsky, Geoffrey Hinton, et al. "Learning multiple layers of features from tiny images". In: (2009).

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems*. Ed. by F. Pereira et al. Vol. 25. Curran Associates, Inc., 2012. URL: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[12] Jan Kukacka, Vladimir Golkov, and Daniel Cremers. "Regularization for Deep Learning: A Taxonomy". In: *CoRR* abs/1710.10686 (2017). arXiv: 1710.10686. URL: http://arxiv.org/abs/1710.10686.

[13] Sihan Li et al. "Demystifying ResNet". In: *CoRR* abs/1611.01186 (2016). arXiv: 1611.01186. URL: http://arxiv.org/abs/1611.01186.

[14] Yehao Li et al. "Contextual Transformer networks for visual recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45.2 (2022), pp. 1489–1500.

[15] Tsung-Yi Lin et al. "Microsoft coco: Common objects in context". In: *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*. Springer. 2014, pp. 740–755.

[16] Nian Liu et al. "Visual saliency Transformer". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2021, pp. 4722–4732.

[17] Xiangyu Liu et al. "L-EfficientUNet: Lightweight End-to-End Monocular Depth Estimation for Mobile Robots". In: *International Conference on Intelligent Robotics and Applications*. Springer. 2023, pp. 394–408.

[18] Yang Liu et al. "A survey of visual Transformers". In: *IEEE Transactions on Neural Networks and Learning Systems* (2023).

[19] Tom O'Malley et al. *KerasTuner*. https://github.com/keras-team/keras-tuner. 2019.

[20] Harry A. Pierson and Michael S. Gashler. "Deep learning in robotics: a review of recent research". In: *Advanced Robotics* 31.16 (2017), pp. 821–835. DOI: 10.1080/01691864.2017.1365009. eprint: https://doi.org/10.1080/01691864.2017.1365009. URL: https://doi.org/10.1080/01691864.2017.1365009.

[21] Ilija Radosavovic et al. "Real-world robot learning with masked visual pre-training". In: *Conference on Robot Learning*. PMLR. 2023, pp. 416–426.

[22] Javier Ruiz-del-Solar, Patricio Loncomilla, and Naiomi Soto. "A Survey on Deep Learning Methods for Robot Vision". In: *CoRR* abs/1803.10862 (2018). arXiv: 1803.10862. URL: http://arxiv.org/abs/1803.10862.

[23] Miguel Saavedra-Ruiz, Sacha Morin, and Liam Paull. "Monocular Robot Navigation with Self-Supervised Pretrained Vision Transformers". In: *2022 19th Conference on Robots and Vision (CRV)*. IEEE. 2022, pp. 197–204.

[24] Claudio Filipi Gonçalves Dos Santos and João Paulo Papa. "Avoiding overfitting: A survey on regularization methods for convolutional neural networks". In: *ACM Computing Surveys (CSUR)* 54.10s (2022), pp. 1–25.

[25] Niko Sünderhauf et al. "The limits and potentials of deep learning for robotics". In: *The International Journal of Robotics Research* 37.4-5 (2018), pp. 405–420. DOI: 10.1177/0278364918770733. eprint: https://doi.org/10.1177/0278364918770733. URL: https://doi.org/10.1177/0278364918770733.

[26] Mingxing Tan and Quoc Le. "Efficientnetv2: Smaller models and faster training". In: *International conference on machine learning*. PMLR. 2021, pp. 10096–10106.

[27] Mingxing Tan and Quoc V. Le. "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks". In: *CoRR* abs/1905.11946 (2019). arXiv: 1905.11946. URL: http://arxiv.org/abs/1905.11946.

[28] Ruihan Yang et al. "Learning Vision-Guided Quadrupedal Locomotion End-to-End with Cross-Modal Transformers". In: *CoRR* abs/2107.03996 (2021). arXiv: 2107.03996. URL: https://arxiv.org/abs/2107.03996.

[29] Yunhua Zhang, Hazel Doughty, and Cees Snoek. "Learning Unseen Modality Interaction". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 54716–54726. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/abb4847bbd60f38b1b7649d26c7a0067-Paper-Conference.pdf.