A purple and black rectangular sign

Description automatically generated

AI ASSIGNMENT 2

(Deep CNNs vs Shallow CNNs)

**B Leotlela**

**34854304**

**Table of Contents**

[**1** **INTRODUCTION** 2](#_Toc148255103)

[**2** **BACKGROUND** 2](#_Toc148255104)

[2.1 An examination of CNNs 2](#_Toc148255105)

[2.2 A comparative examination of Deep and Shallow CNNs 3](#_Toc148255106)

[2.3 Typical CNN architecture 3](#_Toc148255107)

[**3** **MODEL DESCRIPTIONS** 4](#_Toc148255108)

[3.1 Multi-Layered CNN (Model 1) 4](#_Toc148255109)

[3.2 Simplified CNN (Model 2) 5](#_Toc148255110)

[3.3 Comparative Exploration 6](#_Toc148255111)

[**4** **Experimental setup** 6](#_Toc148255112)

[4.1 Dataset 6](#_Toc148255113)

[4.2 Training setup 6](#_Toc148255114)

[**5** **Results and discussion** 7](#_Toc148255115)

[5.1 Results 7](#_Toc148255116)

[5.2 Class-wise performance 9](#_Toc148255117)

[5.3 Depth and Complexity 9](#_Toc148255118)

[5.4 Considerations and Implications 9](#_Toc148255119)

[**6** **Conclusion** 9](#_Toc148255120)

[**7** **References** 10](#_Toc148255121)

**TABLE OF FIGURES**

[Figure 1: Flowchart of Training setup 7](#_Toc148255122)

[Figure 2: Confusion Matrix for Model 1 Figure 3: Confusion Matrix for Model 2 8](#_Toc148255123)

[Figure 4: ROC Curve for Model 1 Figure 5: Roc Curve for Model 2 8](#_Toc148255124)

**TABLE OF TABLES**

[Table 1: Table showing Model 1's architecture 5](#_Toc148255125)

[Table 2: Table showing Model 2's architecture 5](#_Toc148255126)

# **INTRODUCTION**

Image classification, a pivotal domain within computer vision seeks to categorize images into predefined classes based on inherent visual patterns and features. The intricacies of distinguishing diverse and nuanced features within images render it a challenging task, especially when considering variations in scales, rotation, and other transformations (Krizhevsky *et al.*, 2012). Convolutional Neural Networks (CNNs), a subclass of deep learning algorithms, have emerged as a prominent solution, demonstrating exemplary performance in numerous image classification benchmarks (Simonyan & Zisserman, 2014).

The architectural depth of CNNs, symbolizing the number of layers within the network, has been a subject of research and debate within the academic and industrial realms. Deep models, such as VGGNet, have illustrated that increased depts can enhance feature abstraction and consequently, classification accuracy (Simonyan & Zisserman, 2014). However, depth introduces computational and resource complexities, potentially impeding training efficiency and demanding substantial hardware capabilities (He *et al.*, 2016)).

This study seeks to explore CNNs and contrast the performances of deep and shallow CNN models within the context of the CIFAR10 image classification dataset. The endeavour aims to discern the tangible impacts of model depth on classification accuracy, training efficiency, and model generalization, offering insights that might guide practitioners in selecting appropriate model architectures tailored to specific use-cases and computational constraints.

# **BACKGROUND**

### An examination of CNNs

In the realm of computer vision, CNNS have established themselves as a pivotal model, demonstrating adeptness in various tasks like image classification and object detection. CNNs, through their convolution and pooling layers, adeptly learn spatial hierarchies from images, extracting pivotal features that are instrumental for accurate prediction and recognition (Lei *et al.*, 2019). Additionally, a contrast exists between deep and shallow architectures, each bringing forth a unique set of advantages, challenges and applications in various computer vision tasks.

#### Typical CNN architecture

A typical CNN has various types of layers including the convolutional layers, pooling layers, dropout layers and fully connected layers. In addition to this, there are activation functions such as ReLU and Softmax that are used in some CNN layers. These components are briefly discussed below (Lei *et al.*, 2019):

* Convolutional layers extract spatial features from input images through learnable filters, and they produce feature maps highlighting specific patterns.
* Pooling layers reduce the spatial dimensions of feature maps, preserving vital features and decreasing parameters. MaxPooling is commonly used to select maximal values in specified regions
* Dropout layers help mitigate overfitting by randomly deactivating a fraction of neurons during training. They enhance model generalization by preventing co-adaptation of hidden units.
* Fully connected layers perform classification based on learned spatial features. The number and size of these layers influence the model’s complexity and susceptibility to overfitting.
* Activation functions:
* ReLU(Rectified Linear Unit) introduces non-linearity without affecting the receptive fields of convolution layers
* Softmax is used in output layers for multi-class classification, converting outputs into probability distributions.

These layers and activation functions when combined in a CNN, facilitate the extraction, learning, and classification of hierarchical features from input images.

#### The role of fully connected layers in CNNs

In the architecture of CNNs, fully connected (FC) layers stand out for their vital role, especially in classification tasks, by learning high-level representations from the extracted features. While FC layers traditionally find their place towards the end of the network, acting as classifiers based on the features learned through preceding convolutional and pooling layers, their significance is multifaceted and deeply intertwined with the overall network architecture (Basha *et al.*, 2020).

The number of neurons and layers in FC sections can notably influence the model’s capacity to decipher complex representations from the input data. A delicate balance is imperative; while more neurons or additional layers can potentially augment the network’s learning capacity, they also run the risk of overfitting, thereby reducing the model’s generalization to unseen data. The optimal structure, therefore, is often a nuanced equilibrium, harmonizing the model’s learning capacity with its predictive generalization, and always tuned according to the specific task and data at hand (Basha *et al.*, 2020).

### A comparative examination of Deep and Shallow CNNs

#### Deep CNNs

This variation in the architecture of CNNs is characterised by an extensive number of layers. They have etched a notable presence in the domain, demonstrating a proficient ability to learn highly complex features from large datasets and subsequently, achieving remarkable accuracy across a spectrum of tasks (Lei *et al.*, 2019). However, this profound depts also intertwines with escalated computational complexity and memory demands, which can potentially increase the resources required for model training and inference.

#### Shallow CNNs

Conversely, shallow CNNs, while having fewer layers and parameters, emerge as computationally efficient alternatives, particularly apt in scenarios with limited data or computational resources. They have demonstrated a capacity to achieve commendable performance across certain computer vision tasks, especially when the complexity of the task and the size of the available dataset are modest (Lei *et al.*, 2019). However, their ability to extract complex features and handle intricate tasks may be limited compared to their deeper counterparts.

### Typical CNN architecture

A typical CNN has various types of layers including the convolutional layers, pooling layers, dropout layers and fully connected layers. In addition to this, there are activation functions such as ReLU and Softmax that are used in some CNN layers. These components are briefly discussed below:

* Convolutional layers extract spatial features from input images through learnable filters, and they produce feature maps highlighting specific patterns.
* Pooling layers reduce the spatial dimensions of feature maps, preserving vital features and decreasing parameters. MaxPooling is commonly used to select maximal values in specified regions
* Dropout layers help mitigate overfitting by randomly deactivating a fraction of neurons during training. They enhance model generalization by preventing co-adaptation of hidden units.
* Fully connected layers perform classification based on learned spatial features. The number and size of these layers influence the model’s complexity and susceptibility to overfitting.
* Activation functions:
* ReLU(Rectified Linear Unit) introduces non-linearity without affecting the receptive fields of convolution layers
* Softmax is used in output layers for multi-class classification, converting outputs into probability distributions.

These layers and activation functions when combined in a CNN, facilitate the extraction, learning, and classification of hierarchical features from input images.

# **MODEL DESCRIPTIONS**

In this section, we delve into a detailed description of the two CNN models designed for the image classification task on the CIFAR-10 dataset. The dataset, known for its established utility in the AI community, comprises 60000 32x32 colour images spanning 10 different classes (). The models under examination exhibit a slightly varying architectural complexity, serving as a basis to exploure the impact of model depth and layer configurations on performance.

### Multi-Layered CNN (Model 1)

The first model, which we will refer to as the Multi-Layered CNN, leans towards a deeper architecture, somewhat inspired by the VGGNet model. This model incorporates multiple convolutional layers, each interspersed to mitigate overfitting. The inclusion of numerous layers and nodes is aimed at enabling the model to learn a rich hierarchy of features from the CIFAR-10 dataset.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Layer No. | Layer Type | Filters/Units | Kernel Size | Stride | Padding | Activation | Regularization | Dropout |
| 1 | Convolutional | 64 | 3x3 | 1x1 | same | ReLU | L2 (Weight Decay) | No |
| 2 | Dropout | - | - | - | - | - | - | Yes |
| 3 | Convolutional | 64 | 3x3 | 1x1 | same | ReLU | L2 (Weight Decay) | No |
| 4 | Max Pooling | - | 2x2 | 2x2 | - | - | - | No |
| 5 | Convolutional | 128 | 3x3 | 1x1 | same | ReLU | L2 (Weight Decay) | No |
| 6 | Dropout | - | - | - | - | - | - | Yes |
| 7 | Convolutional | 128 | 3x3 | 1x1 | same | ReLU | L2 (Weight Decay) | No |
| 8 | Max Pooling | - | 2x2 | 2x2 | - | - | - | No |
| 9-11 | Convolutional | 256 | 3x3 | 1x1 | same | ReLU | L2 (Weight Decay) | No |
| 12 | Dropout | - | - | - | - | - | - | Yes |
| 13-14 | Convolutional | 256 | 3x3 | 1x1 | same | ReLU | L2 (Weight Decay) | No |
| 15 | Max Pooling | - | 2x2 | 2x2 | - | - | - | No |
| 16 | Flatten | - | - | - | - | - | - | No |
| 17 | Fully Connected | 512 | - | - | - | ReLU | L2 (Weight Decay) | No |
| 18 | Dropout | - | - | - | - |  |  | Yes |
| 19 | Fully Connected | 10 | - | - | - | Softmax |  | No |

Table 1: Table showing Model 1's architecture

**Architecture**

* Multiple convolutional blocks, each consisting of convolutional layers followed by activation and batch normalization, designed to extract hierarchical features from the input images.
* Dropout layers interspersed to mitigate overfitting and foster the training of a robust model.
* Three MaxPooling layers to progressively reduce special dimensions.
* Dense layers at the end to facilitate the learning of non-linear combinations of high-level features.
* A final output layer with softmax activation to classify the input images into one of the 10 classes.

**Noteworthy Aspects**

* The model makes use of L2 regularization and batch normalization
* Employs dropout layers to introduce regularization and reduce overfitting.
* The model is comparatively deep, potentially enabling it to learn more complex and hierarchical features from the data.

### Simplified CNN (Model 2)

The second model, herein referred to as the Simplified CNN, is architecture with a ore conservative approach, featuring fewer layers than its counterpart. This model consists of three convolutional blocks, each followed by a max-pooling layer, and concludes with a couple of dense layers for classification.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Layer No. | Layer type | Filters/Units | Kernel Size | Stride | Padding | Activation | Regularization | Dropout |
| 1 | Convolutional | 32 | 3x3 | 1x1 | - | ReLU | - | No |
| 2 | Max Pooling | - | 2x2 | 2x2 | - | - | - | No |
| 3 | Convolutional | 64 | 3x3 | 1x1 | - | ReLU | - | No |
| 4 | Max Pooling | - | 2x2 | 2x2 | - | - | - | No |
| 5 | Convolutional | 128 | 3x3 | 1x1 | - | ReLU | - | No |
| 6 | Max Pooling | - | 2x2 | 2x2 | - | - | - | No |
| 7 | Flatten | - | - | - | - | - | - | No |
| 8 | Fully Connected | 64 | - | - | - | ReLU | - | No |
| 9 | Fully Connected | 10 | - | - | - | Softmax | - | No |

Table 2: Table showing Model 2's architecture

**Architecture**

* Three convolutional blocks, each comprising a convolutional layer followed by an activation layer, intended to extract fundamental features from the images.
* MaxPooling layers following each convolutional block to downsample the feature maps and reduce computational load.
* A Flatten layer to convert the 2D feature maps to a 1D vector.
* A dense layer with ReLu activation to learn non-linear mappings.
* A final Dense layer with softmax activation for classification into 10 categories.

**Noteworthy Aspects**

* The model is relatively shallow, which could make it computationally efficient and potentially less prone to overfitting, albeit at the risk of capturing less intricate features.
* The model does not employ dropout or batch normalization, providing a basis to assess how these augmentations impact model performance.

### Comparative Exploration

A pivotal question this investigation seeks to answer pertains to the trade-off between architectural complexity and predictive performance. In subsequent sections, we will delve into experiments to critically evaluate and contrast the performance of these two models on the CIFAR-10 dataset, exploring aspects such as accuracy, loss, training time, and potential overfitting, to glean insights into the implications of model depts and layer configurations in CNNs. We have outlined the architecture of two distinct CNN models. The subsequent sections of your report could delve into the implementation details, training methodology, and an evaluation and comparison of the models based on various metrics and visualisations.

# **Experimental setup**

### Dataset

To derive the empirical observations presented in this report, experiments were conducted using the CIFAR-10 dataset, which is widely recognized for its application in small image classification across ten distinct categories, each with 6,000 images. Each image in the dataset has dimensions of 32x32x3. For training the neural networks, the training set of the CIFAR-10 dataset, encompassing 50,000 images was employed, while the remaining 10,000 images were used to validate the performance of the models. Fig.1 showcases a few sample images from the CIFAR-10 dataset.

### Training setup

The training process that was followed for both methods is depicted in the flowchart below:

A diagram of a model

Description automatically generated

Figure 1: Flowchart of Training setup

# **Results and discussion**

### Results

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Model 1 (Deep CNN) | | | Model 2 (Shallow CNN) | | |
|  | Precision | Recall | F1-score | Precision | Recall | F1-score |
| Class1 | 75% | 76% | 76% | 73% | 81% | 77% |
| Class2 | 90% | 84% | 87% | 78% | 89% | 83% |
| Class3 | 72% | 52% | 61% | 71% | 51% | 59% |
| Class4 | 51% | 66% | 58% | 58% | 47% | 52% |
| Class5 | 73% | 66% | 69% | 72% | 59% | 65% |
| Class6 | 57% | 71% | 63% | 63% | 59% | 61% |
| Class7 | 77% | 80% | 79% | 62% | 89% | 73% |
| Class8 | 85% | 65% | 74% | 67% | 83% | 74% |
| Class9 | 80% | 90% | 85% | 87% | 76% | 81% |
| Class10 | 84% | 81% | 83% | 83% | 79% | 81% |
| Accuracy |  |  | 73% |  |  | 71% |
| Weighted average | 75% | 73% | 73% | 71% | 71% | 71% |

Table 3:Table indicating the classification reports of the models

A graph of confusion matrix

Description automatically generated A graph of confusion matrix

Description automatically generated

Figure 2: Confusion Matrix for Model 1 Figure 3: Confusion Matrix for Model 2

A graph of a function

Description automatically generated with medium confidence A graph of a function

Description automatically generated with medium confidence

Figure 4: ROC Curve for Model 1 Figure 5: Roc Curve for Model 2

These classification reports reveal insightful details about the predictive performance of each model across the ten classes. There is a noticeable distinction in performance that can be observed between the models, albeit not substantially disparate. The Deeper model achieve an overall accuracy and average F1-score of 73%, slightly edging out the simpler CNN which accomplished respective scores of 71%. This marginal superiority of the deeper model (model 1) aligns with the expectation that deeper networks given their capacity to learn more complex features might perform better in certain tasks. The ROC curves and Confusion Matrices for both models shows that the models perform exceptionally well in classifying the images.

### Class-wise performance

Intriguingly, both models demonstrated higher F1-scores for Class 2 and Class 9, signifying a protentially easier classificiation or distinctive features within these classes. However, disparities in performance across other classes reveal the unique learning mechanisims and feture extraction capabilities inherent to each model. For instance, Model 1 excelled in identifying Class7 and Class8, while the simpler model achieved commendable scores for Class10 and Class7.

### Depth and Complexity

Model 1, with its deeper architecture and additional regularization mechanisms, such as dropout, might possess an enhanced ability to learn intricate patterns within the data. However, the simpler model, with its concise architecture and reduced parameter space, demonstrated competitive performance, therby substantiating the notion that shollow networks can indeed be proficient, particularly when computational resources or data are constrained

### Considerations and Implications

Computational efficiency: Despite the slightly superior performance of Model 1, it’s imperative to consider the computational resources and training time. The simpler model, with commendable accuracy, might be prefered in scenarious demanding computational efficiency.

Overfitting: Model 1 , with its depts and complexity, might be more prone to overfitting, particularly with limited data, despite the incorporation of dropout layers. The simpler model might offer a more genralized performance with reduces susceptibility to overfitting.

Interpretability: Simpler models often lend themselves to enhanced interpretability, a crucial factor in certain applications where model understanding and trust are paramount.

# **Conclusion**

Both models exhibited proficient performance in classifying images across the 10 classes in the CIFAR-10 dataset. The deeper model (Model 1) marginally outperformed the simpler CNN, yet the latter, with its computational efficiency and reduced complexity, illustrated significant potential. This analysis underscores the importance of selecting model architectures that align with the specific demands and applications of the applications at hand. The trade-off between model complexity and predictive performance reains a pivotal consideration, advocating for nuanced, application-specific model selection.

The network depth, while being influential, is intertwined with other pivotal factors such as the complexity and number of filters, activation functions, and applied regularization techniques in determining the model’s performance (Lei *et al.*, 2019) . Thus, the journey towards selecting an optimal architecture – deep or shallow – is often dictated by a confluence of factors including the specific task, available data, and computational resources, necessitating a tailored approach that balances depth with computational efficiency and model performance.

# **References**

Basha, S.H.S., Dubey, S.R., Pulabaigari, V. & Mukherjee, S. 2020. Impact of fully connected layers on performance of convolutional neural networks for image classification. *Neurocomputing*, 378:112-119. <https://www.sciencedirect.com/science/article/pii/S0925231219313803> <https://doi.org/10.1016/j.neucom.2019.10.008>

He, K., Zhang, X., Ren, S. & Sun, J. 2016. Deep Residual Learning for Image Recognition*.* In. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 770-778.

Krizhevsky, A., Sutskever, I. & Hinton, G. 2012. ImageNet Classification with Deep Convolutional Neural Networks. *Neural Information Processing Systems*, 25, 10.1145/3065386

Lei, F., Liu, X., Dai, Q. & Ling, B.W.-K. 2019. Shallow convolutional neural network for image classification. *SN Applied Sciences*, 2(1):97. <https://doi.org/10.1007/s42452-019-1903-4> 10.1007/s42452-019-1903-4

Simonyan, K. & Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv 1409.1556*,