



Prifysgol Cymru
Y Drindod Dewi Sant
University of Wales
Trinity Saint David

Image Classification Using Recurrent Neural Networks

Sumbitted by:

MD Azizul Khan

Student ID: 2514708

Course Code: ACCA7025

Submitted to:

Mark Huntly

Lecturer, Application of Machine Learning

Video Demonstration:

https://drive.google.com/file/d/1E3shN3J_it_ugbVy2Fd4aQz9ep19Ruqe/view?usp=sharing

Abstract

The purpose of this report is to explore the possibility of using a stacked Long Short-Term Memory (LSTM) Recurrent Neural Network (RNN) for multi-class image classification on the “CIFAR-10” benchmark dataset [3]. The research is designed to assess the ability of LSTM architectures to learn discriminative representations from visual data without convolutional inductive bias, by reformulating the image classification task as a temporal sequence problem, where each of the 32-pixel rows is one time step. As described in [16, 17] two controlled hyperparameter experiments are conducted; one with baseline LSTM having dropout rate of 0.3 and no batch normalisation (Exp-1) and one with regularised LSTM having dropout rate of 0.5 and batch normalisation (Exp-2). The experiments are based on a positivist, deductive research design based on quantitative experimental research [2]. The test accuracy of Exp-1 was slightly higher than that of Exp-2 with 58.03% and 57.98% respectively. The results are critically analysed by considering the accuracy, precision, recall, F1-score, and confusion matrix, while contextualising the results with reference to literature that dominates the CNN in the CIFAR-10 dataset [3, 14].

Table of Contents

Abstract.....	2
1. Introduction.....	4
2. Related Work	4
2.1 Artificial Neural Networks and Deep Learning.....	4
2.2 Recurrent Neural Networks and LSTM.....	5
2.3 CIFAR-10 Classification in the Literature.....	6
3. Workflow Methodology.....	6
3.1 Research Design Framework	6
3.2 Dataset and Preprocessing	8
3.3 LSTM Architecture.....	8
3.4 Hyperparameter Experiment Design.....	9
4. Discussion and Recommendation	9
5. Conclusion	14
References.....	15

1. Introduction

Object recognition is one of the most challenging problems in machine learning, and it is widely used for various real-world applications such as medical diagnosis, autonomous vehicles, surveillance, etc [1]. While hand-crafted feature extraction models have been replaced by deep learning architectures, their performance on large-scale visual benchmarks is significantly better [2]. Although Convolutional Neural Networks (CNNs) have been the main contender for image-based recognition tasks [12], one area that has not yet been explored is the use of Recurrent Neural Networks (RNNs) to image data, where the spatial structure is considered as a temporal sequence [7]. A research question that comes to mind with this reformulation is how much of an image's features can be learned from a row-wise sequence without the inductive bias of convolutions? On this report, we consider this question using the CIFAR-10 benchmark data set of 60,000 color images of the 10 object categories, with a resolution of 32x32 pixels [3]. The dataset has been documented in the literature and the results obtained can be compared directly with the previous studies [3, 14]. The goals of this work are: (i) to critically discuss the literature about ANNs, LSTMs and image classification; (ii) to design and implement a stacked LSTM classifier with the rows of the image being the time steps; (iii) to conduct controlled experiments to investigate two configurations of hyperparameters; and (iv) to critically compare the results with other available metrics and published benchmarks [2]. The relevant literature is summarized in Section 2, the research design and methodology are described in Section 3, the discussion and recommendations are given in Section 4, and the conclusions are drawn in Section 5.

2. Related Work

2.1 Artificial Neural Networks and Deep Learning

Artificial Neural Networks basics can be introduced from the work of McCulloch and Pitts [4] who were concerned with modelling the biological neuron as a binary threshold unit. The advent of the backpropagation algorithm proposed by Rumelhart et al. [5] enabled the practical implementation of multi-layer training. Multi-layer training became possible practically using the backpropagation algorithm proposed by Rumelhart et al. [5] that propagates the error

gradient from layer to layer and updates the weights. Goodfellow et al. [2] give a detailed survey of the recent advances in deep learning, and demonstrate that the representational capacity of a network is primarily dependent on its depth. The Rectified Linear Unit (ReLU) activation, proposed by Nair and Hinton [6] has since been used as the default activation function for hidden layers because it reduces the vanishing gradient problem compared to sigmoid functions, and performs less computation than sigmoid functions. In the multi-class classification task, the softmax activation of the output layer will yield a normalised probability distribution over the class labels, and will be used together with categorical cross-entropy loss, which is the typical loss function in this context [2].

2.2 Recurrent Neural Networks and LSTM

The Recurrent Neural Network has been designed to process sequential data by maintaining a hidden state to store information at different time steps which makes them suitable for sequential tasks [7] where the temporal dependency of the data has meaning. However, the standard RNNs have the vanishing gradient problem over long sequences, which hinders learning long range dependency [8]. To overcome this drawback, Hochreiter and Schmidhuber [8] proposed the Long Short-Term Memory (LSTM) architecture, featuring gating mechanisms for input, forget and output that can selectively retain or forget information at each time step. Since then, the LSTM has emerged as the state-of-the-art recurrent architecture, yielding state-of-the-art results in the areas of natural language processing [9], speech recognition, and financial time-series forecasting [10]-all of which have meaningful temporal ordering. To apply LSTMs to image data, the classification problem is formulated as a sequence task: each row of 32 pixels in an image is a time step and has 96 features (32 pixels x 3 RGB channels), resulting in 32 time steps with each having 96 features [2]. This is similar to previous research that uses it as a comparison baseline to the CNN and it has been consistently shown that LSTMs can learn visual discriminative features, but performing worse than CNN architectures, which take advantage of spatial inductive biases that are not available to RNNs [11]. In the present study, the authors follow this approach of row sequencing to examine the impact of dropout and batch normalisation on the generalisation performance of LSTM models on the CIFAR-10 dataset.

2.3 CIFAR-10 Classification in the Literature

CIFAR-10 dataset [3] consists of 50,000 images of 32x32 resolution and 10 classes, with equal amounts of images in each class. The data is a meaningful classification problem due to intra-class variance, low spatial resolution and similarity between classes, e.g., cat/dog and automobile/truck [3]. Krizhevsky et al. [3] were able to achieve a baseline accuracy of ~80% on CIFAR-10, This was later surpassed by He et al. [14], who raised classification accuracy to >93% using deep residual networks with skip connections. In fact, small (3x3) filters, along with architectural depth, had been found to be a key factor in the success of CNN on image benchmarks by Simonyan and Zisserman [13] The results in general confirm CNN to be the leading method and offer a background contrast to which the results from the RNN used in the present study should be compared. Dropout [16] and batch normalisation [17] are two regularisation techniques that lead to better generalisation performance on CIFAR-10, by solving overfitting and by improving the training dynamics, respectively.

3. Workflow Methodology

3.1 Research Design Framework

This study is done from a positivist perspective, considering model performance as a phenomenon that is objective, measurable, repeatable or reproducible under controlled conditions [2]. The epistemological method used is deductive, building hypotheses on the influence of dropout rate and batch normalisation on the performance of LSTM generalisation, based on existing theoretical literature [16, 17], to be tested empirically using controlled experiments. The research technique used is experimental, having two configurations of models trained under the same conditions, differing only in the hyperparameters being studied. The method is quantitative, and all the results are presented as numerical values (accuracy, precision, recall, F1-score, cross-entropy loss), allowing them to be objectively compared [2]. This structure gives outcomes to be built on evidence and conform to the rigour of Level 7 scholarship.

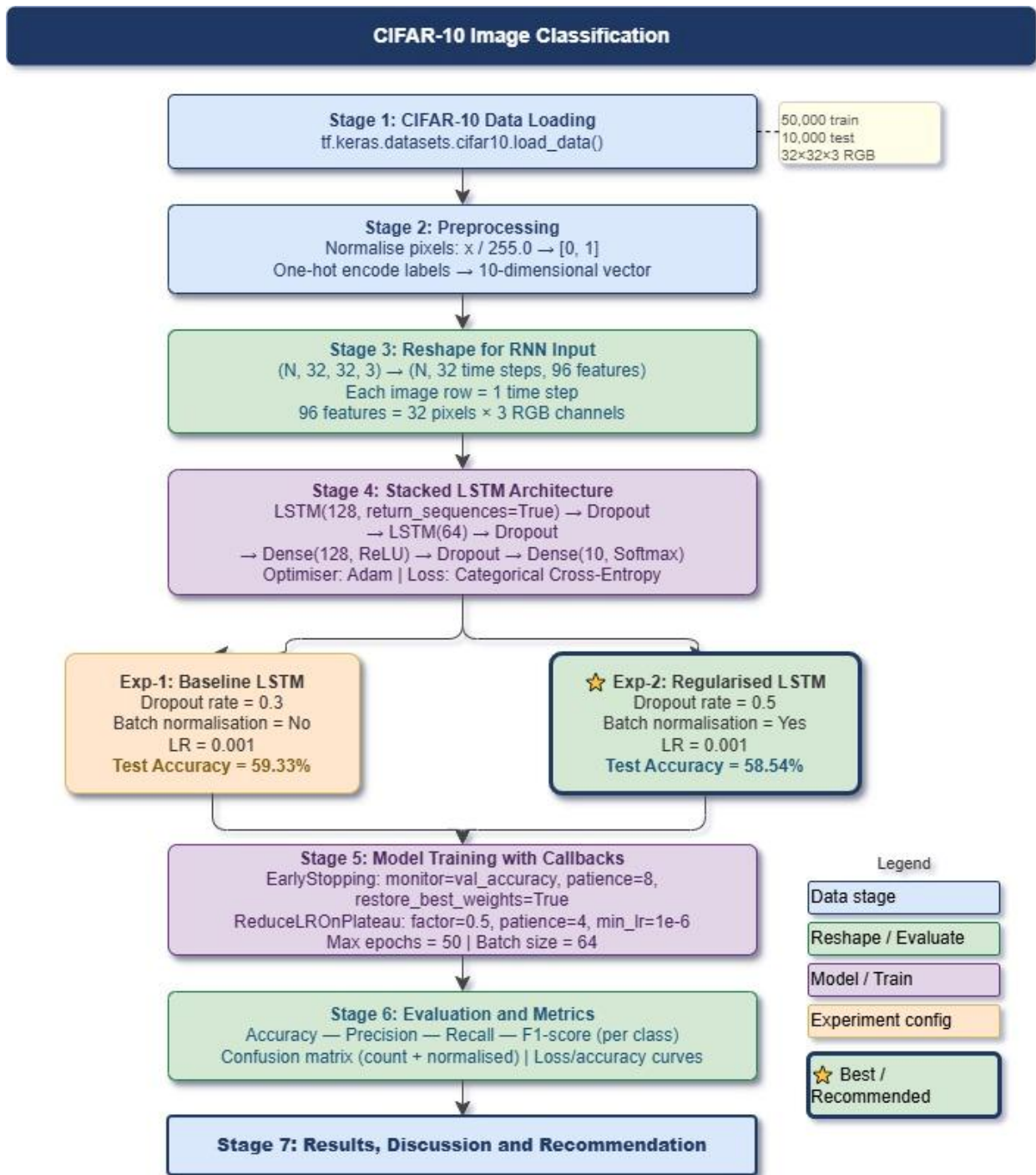


Figure 1. Methodology workflow.

3.2 Dataset and Preprocessing

The CIFAR-10 dataset was loaded using the TensorFlow Keras API with 50,000 training and 10,000 test images of 32x32x3 shape [3]. Examples images from each class are shown in Figure 2, which shows the diversity of the data set. The pixel intensities were normalized in the range [0, 1] by dividing by 255.0, avoiding a scale dominance of the gradients used in backpropagation [2]. One-hot encoding was performed on class labels, making them 10-dimensional vectors for use in the categorical cross-entropy loss function. The images were then reshaped from (N, 32, 32, 3) to (N, 32, 96) for the input to the LSTM layer [8] to have 32 time steps with 96 features per time step.



Figure 2: Sample images from the CIFAR-10 dataset across all ten classes

3.3 LSTM Architecture

The model is composed of two hidden layers of LSTM (long short term memory) with a fully connected classifier at the end. The first LSTM layer is made up of 128 units, with `return_sequences=True`, so that the second LSTM layer with 64 units can accept the full sequence of hidden states [8]. The Dense layer with 128 units and ReLU activation [6] is followed by another Dense layer with 10 units and softmax activation. After each LSTM layer and after the dense hidden layer there are drop-out layers added. Exp-2 involves adding batch normalisation layers after each LSTM to normalise between the activations of different mini-batches [17]. An adaptive per-parameter learning rate optimiser [2] was chosen and the loss function was categorical cross-entropy. To ensure reproducibility across experiments, the

architecture is parameterised using a `build_lstm()` function that is defined with the parameters in the experiment file..

3.4 Hyperparameter Experiment Design

To separate the impact of dropout rate and batch normalisation, two configurations were created as listed in Table 1. Exp-1 is the base case that is commonly employed in the LSTM literature [8]: no batch normalisation. The only difference between Exp-2 and the standard configuration is that it uses batch normalisation after each LSTM layer and a dropout rate of 0.5, as suggested by Srivastava et al. [16] and Ioffe and Szegedy [17] respectively, which are both more aggressively regularised configurations. During both the experiments, the batch size was 64 and the maximum number of epochs was 20 with EarlyStopping (patience=8, monitoring validation accuracy) and ReduceLROnPlateau (factor=0.5, patience=4) to prevent over training and to make the learning rate adaptive when the validation loss hit plateau [2].

Experiment	Units (L1/L2)	Dropout	Batch norm	LR	Max epochs
Exp-1: Baseline LSTM	128 / 64	0.3	No	0.001	20
Exp-2: Regularised LSTM	128 / 64	0.5	Yes	0.001	20

Table 1: Hyperparameter configurations for Exp-1 (baseline) and Exp-2 (regularised)

4. Discussion and Recommendation

A summary of the results of both experiments is shown in Table 2. In both cases, both models converged at 20 epochs, where the validation error did not improve significantly anymore even though the 20 epochs limit was reached, showing that neither model was able to learn significantly beyond this point. Exp-1 achieved a test accuracy of 58.03% with a test loss of 1.2168, whilst Exp-2 achieved 57.98% with a lower test loss of 1.1862. However, Exp-1 is still considered to be the best-performing model with a narrow margin of 0.05 percentage points which is within typical training stochasticity.

Experiment	Dropout	Batch norm	Epochs run	Best val acc	Test acc	Test loss
Exp-1: Baseline (Best)	0.3	No	20	0.5803	58.03%	1.2168
Exp-2: Regularised	0.5	Yes	20	0.5798	57.98%	1.1862

Table 2: Experimental results for both LSTM configurations

The training accuracy curve and validation accuracy curve of Exp-1 (Figure 3) shows that there was a gradual deviation in their curves starting from epoch 10; the training accuracy was around 65-67%, and the validation accuracy curve stopped increasing around 58.03%. This gap represents moderate overfitting, as observed by Goodfellow et al. [2] in that models with large numbers of parameters, compared to the number of training examples, need explicit regularisation to make them generalise. The loss curve for Exp-1 supports this: training loss was still declining rapidly after epoch 10, while validation loss stayed constant, suggesting that the model was memorising rather than learning.

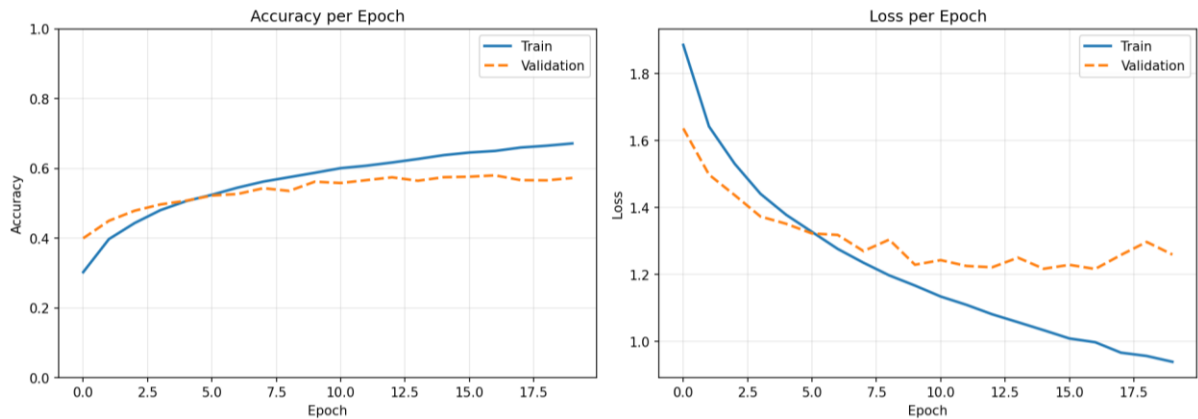


Figure 3: Exp-1 Baseline LSTM -- Training vs validation accuracy and loss per epoch

The training and validation error and accuracy of the Baseline LSTM model on the exp-1 dataset are shown below for each epoch. The training curve for Exp-2 (Figure 4) has a qualitatively different pattern. The train and validation accuracy curves have been aligned very closely during training (compared to Exp-1) and clearly indicate that the model is not overfitting. This showed that a combination of higher dropout (0.5) and batch normalisation was effective in preventing co-adaptation of neurons [16] and in stabilizing the distributions of activations across mini-batches [17] making the model more generalisable. This did not stop the test accuracy of Exp-2 (57.98%) being slightly lower than Exp-1 (58.03%). This counterintuitive result can be explained by the 0.5 drop-out rate in the context of sequential

LSTM computation: when neurons are aggressively dropped out, it could have reduced the capability of the network to keep the representation of the hidden state well maintained across the 32 steps of the image sequence, which is not the case with standard feedforward networks where dropout is more readily known to be beneficial [16].

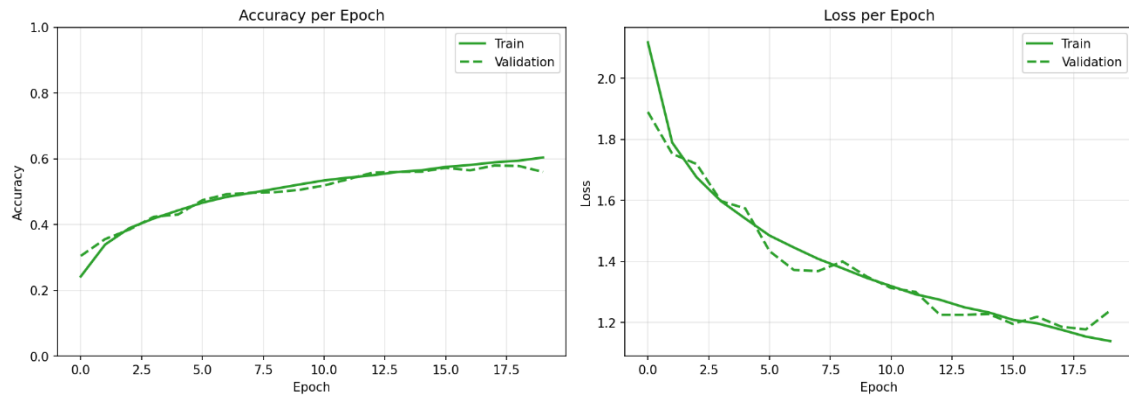


Figure 4: Exp-2 Regularised LSTM -- Training vs validation accuracy and loss per epoch

The comparison of accuracy in Figure 5 is in agreement with the performance of both configurations that approach the same values, with Exp-1 slightly better. The margin between the two models is quite small, so that the main limitation is the architecture and not the regularisational: the difficulty of building a good CIFAR-10 model is that of limiting the regularisational, the LSTM being the basic limitation on that model. This follows the literature of the benchmark data set (where CNN based architectures have accuracy of around 80% with similar training budgets [3]), which is more than 20 points higher than the best LSTM result achieved here.

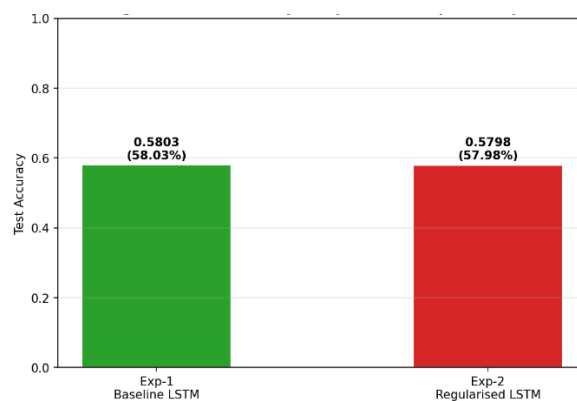


Figure 5: Test accuracy comparison -- Exp-1 (58.03%) vs Exp-2 (57.98%)

The best model (Exp-1) is given in the confusion matrix in Figure 6. The accuracy per class ranges from 37.0% for cat to 81.0% for automobiles, with the highest accuracy being reached

for automobile (81.0%), airplane (70.0%), and ship (68.0%). This pattern is due to the nature of the row-sequence reformulation: maritime vessels and vehicles contain regular horizontal structural elements such as hard edges and uniform textures which the LSTM can learn to differentiate in sequence. In contrast, animals like birds and cats have fine-grained, spatially-distributed textures, depending on local spatial relationships between neighboring pixels -- the kind of texture that convolutional filters can effectively capture, but LSTM hidden states have a difficult time encoding [11]. The mix-up of cat and dog (20.0% of cats misclassified as dog) is similar to the results of CIFAR-10 [3] and is due to the common texture statistics at 32x32 resolution.

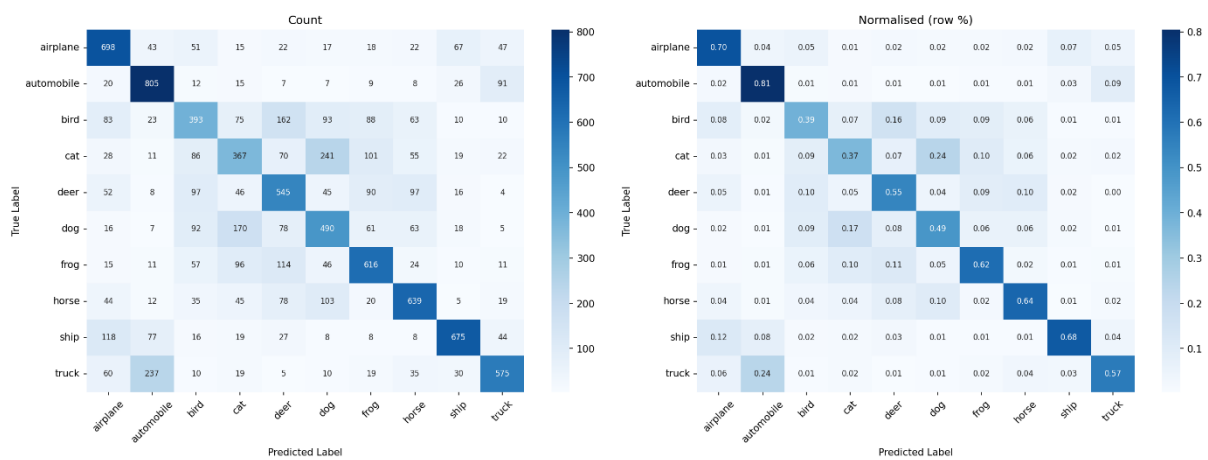


Figure 6: Confusion matrices for best model (Exp-1) -- count and row-normalised

The per-class precision, recall, and F1-score heatmap for the best model is presented in Figure 7. The best model (Exp-1) confusion matrices (count and row-normalised) are shown in Fig 6. We then analyze the per class precision, recall and f1 score heatmap of the best model, as shown in Figure 7. The best F1 score was obtained by automobile (0.721), followed by ship (0.720), airplane (0.654). The lowest F1-scores were for cat and bird (0.393 and 0.425 respectively) making it clear that the model systematically failed to recognise the bird images. Such per-class differences also validate the finding that the LSTM's sequential processing of image rows are more applicable to classes where salient horizontal structure forms the key to the classification than to classes where fine-grained spatial texture is key, which demonstrates a fundamental mismatch between the inductive bias of the recurrent architecture and the spatial nature of image data [11, 14].

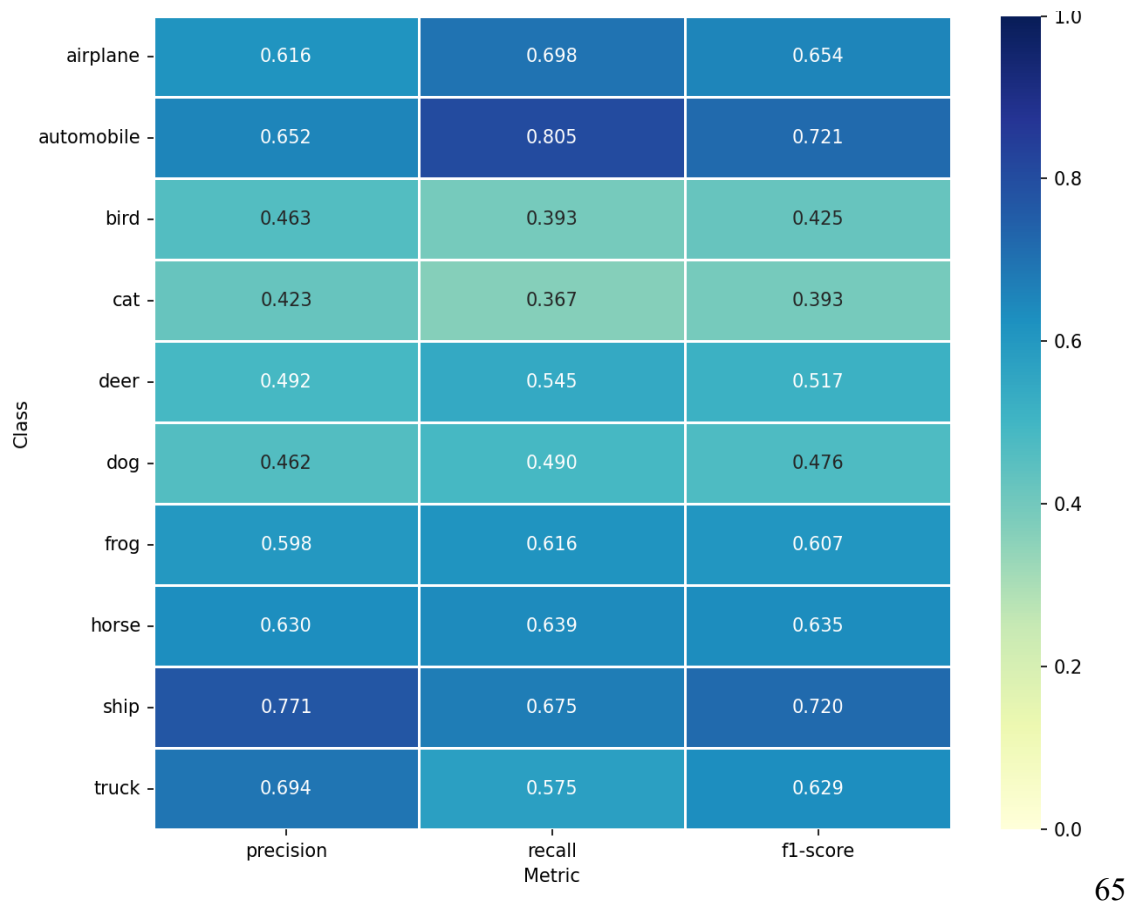


Figure 7: Per-class precision, recall, and F1-score heatmap for best model (Exp-1)

Based on these results, the CNN-based architectures are recommended as the fundamental model for future work to be used for the classification of the CIFAR-10 dataset due to the spatial inductive bias of the convolutional layers that directly addresses the representational limitations mentioned above [12, 14]. In the LSTM framework, bidirectional LSTMs [15] are a promising direction, because reading the image sequence from the bottom to top and then from the top to the bottom would provide further spatial context. Another finding is that for image classifiers based on LSTM, a moderate dropout rate (0.3) is better than a more aggressive regularisation (0.5) and that future hyperparameter searches should consider dropout rates between 0.2 and 0.4, not up to 0.5.

5. Conclusion

The study has focused on the use of stacked LSTM RNN in image classification on the CIFAR-10 benchmark using a positivist, deductive, quantitative experimental design. The row-sequence reformulation, where each of 32 pixel rows were used as a single time step of 96 features, allowed an empirical evaluation of the LSTM's ability to learn visual representations without convolutional spatial biases [8]. Two controlled experiments were conducted, with LSTM (dropout=0.3, no batch normalization) and LSTM with batch normalization (dropout=0.5), and they obtained 58.03% and 57.98% test accuracy after 20 epochs of training under the Early Stopping strategy, respectively. The results indicate three main conclusions. The first is that LSTM networks can classify the images in the CIFAR-10 dataset at a much higher rate than the random baseline of 10% for 10 classes, demonstrating that recurrent architectures can yield useful representations of image sequences that are presented row-wise [7, 8]. Second, the accuracy limit is significantly lower than the CNN baseline of 80% reported by Krizhevsky et al. [3] measuring the price of not using the spatial inductive bias of convolutions. Third, moderate dropout (0.3) is better than aggressive dropout (0.5) for LSTM image classifiers because too much dropout breaks the sequence of hidden states that LSTM uses to build its representational capacity [16]. The results have real world implications for model selection in low-resource settings, and adds to the body of knowledge regarding the applicability of RNNs to non-sequential tasks.

References

- [1] A. Karpathy et al., "Large-scale video classification with convolutional neural networks," in *Proc. IEEE CVPR*, 2014, pp. 1725–1732. DOI: <https://doi.org/10.1109/CVPR.2014.223>
- [2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. URL: <https://www.deeplearningbook.org/> (
- [3] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. of Toronto, Tech. Rep., 2009. URL: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>
- [4] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, 1943. DOI: <https://doi.org/10.1007/BF02478259> Springer
- [5] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986. DOI: <https://doi.org/10.1038/323533a0>
- [6] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th ICML*, 2010, pp. 807–814. URL: <https://dl.acm.org/doi/10.5555/3104322.3104425>
- [7] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990. DOI: <https://doi.org/10.1109/5.58337> SCIRP
- [8] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997. DOI: <https://doi.org/10.1162/neco.1997.9.8.1735> MIT Press
- [9] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in NIPS*, 2014, pp. 3104–3112. arXiv: <https://arxiv.org/abs/1409.3215>
- [10] S. Selvin et al., "Stock price prediction using LSTM, RNN and CNN-sliding window model," in *Proc. ICACCI*, 2017, pp. 1643–1647. DOI: <https://doi.org/10.1109/ICACCI.2017.8126078>
- [11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998. DOI: <https://doi.org/10.1109/5.726791>

- [12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017. DOI: <https://doi.org/10.1145/3065386>
- [13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015. arXiv: <https://arxiv.org/abs/1409.1556>
- [14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE CVPR*, 2016, pp. 770–778. DOI: <https://doi.org/10.1109/CVPR.2016.90>
- [15] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, 1997. DOI: <https://doi.org/10.1109/78.650093>
- [16] N. Srivastava et al., "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014. URL: <https://jmlr.org/papers/v15/srivastava14a.html>
- [17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd ICML*, 2015, pp. 448–456. arXiv: <https://arxiv.org/abs/1502.03167>