COP528

AI and Applied Machine Learning

Professor: Dr Hui Fang

Student ID: F220378

Deep Learning for Image Classification

I. Introduction

In the ever-expanding realm of artificial intelligence and machine learning, image classification has emerged as a pivotal area of research and innovation, serving as the bedrock for an array of groundbreaking applications across various fields, including autonomous vehicles, medical diagnostics, surveillance systems, robotics, and beyond. Deep learning, with a particular emphasis on Convolutional Neural Networks (CNNs), has been the driving force behind the relentless pursuit of state-of-the-art advancements in image classification. As preeminent researchers and practitioners endeavour to develop increasingly sophisticated models, the exploration of diverse architectures, training techniques, and optimisation strategies has taken centre stage.

In this comprehensive study, we embark on an incisive comparative analysis of an assortment of deep learning models and techniques, harnessing the prowess of both pre-trained and custom-built architectures to illuminate their respective merits, limitations, and generalisation capabilities. Our investigation encompasses a triad of meticulously designed experiments that delve into the performance of a pre-trained ResNet50 model, an EfficientNetB0 model trained from the ground up, and an EfficientNetB0 model fine-tuned using the power of transfer learning.

This report is systematised as follows: Section II delineates the dataset employed in this study, expounding upon its composition, classes, and pre-processing steps. Section III elucidates the methodology, encompassing the deep learning models, training strategies, and evaluation metrics employed. Section IV unveils a comprehensive analysis of the results, engaging in a profound discourse on the performance and implications of each experiment. Lastly, Section V offers concluding insights, accentuating key discoveries and potential avenues for future research.

By meticulously dissecting these models and techniques, we aspire to impart invaluable perspicuity into the intricate realm of deep learning for image classification and make a lasting contribution to the ongoing discourse that shapes the future of this revolutionary field.

Our findings not only accentuate the exceptional benefits of fine-tuning pre-trained models via transfer learning but also unveil the limitations and opportunities inherent in alternative approaches. In so doing, we strive to provide indispensable guidance for esteemed researchers and practitioners seeking to optimise their image classification techniques in real-world applications. As you delve into this report, we invite you to join us in exploring the trailblazing deep learning-based image classification with the aim of advancing knowledge and fostering new findings in this transformative domain.

II. Data and Preliminary Analysis

In this section, we have undertaken a meticulous examination of the dataset for our image classification project employing CNNs. This comprehensive analysis focuses on understanding the dataset's class distribution and image file size characteristics, which are paramount in selecting appropriate pre-processing strategies and designing sophisticated model architectures.

Utilising state-of-the-art data analysis and visualisation techniques, we have leveraged libraries such as Tensorflow, Pandas, Matplotlib, and Seaborn to gain valuable insights into the dataset. The ImageNet class index has been mapped to human-readable labels using a JSON file from Kaggle, enabling an intuitive interpretation of the data.

A. *Analysis of Class Distribution*

This section provides a detailed account of the class distribution counts and percentages, supplemented by informative visualisations.

The dataset encompasses 10 distinct classes, with the number of images per class in the training dataset ranging from 858 to 993 and in the validation dataset from 357 to 419. To identify any potential imbalances, we conducted a thorough investigation of class distribution counts and percentages and visualised the results.

Class Label	Training Counts	Validation Counts
tench	963	387
english_springer	955	395
cassette_player	993	357
chain_saw	858	386
church	941	409
french_horn	956	394
garbage_truck	961	389
gas_pump	931	419
golf_ball	951	399
parachute	960	390

Table 1: Training and Validation Datasets Class Distribution - Counts

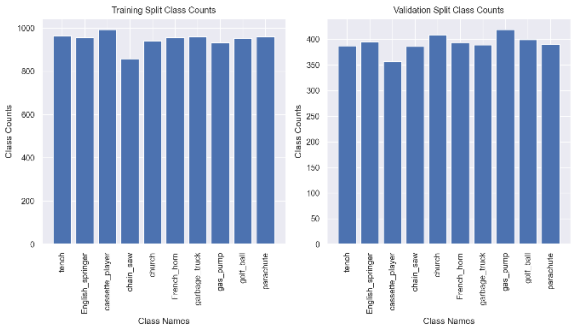

Figure 1: Training and Validation Datasets Class Distribution - Counts

Our findings reveal a relatively even class distribution, with the training dataset percentages varying from 9.06% to 10.49% and the validation dataset percentages fluctuating between 9.10% and 10.68%. This equitable distribution ensures that our model will not be biased towards any particular class during the training process.

Figure 2: Training and Validation Datasets Class Distribution - Percentages

B. Assessment of Image File Size Distribution

This section presents a comprehensive and sophisticated analysis of image file size distribution, elucidating essential insights to guide the development of our image classification model and architecture.

We have meticulously analysed the distribution of image file sizes within each class in the training dataset. This assessment is vital for informing our pre-processing strategies and model architectures, as variations in file sizes may impact the model's performance and generalisability.

Our investigation unveils significant disparities in image file size statistics across different classes. For instance, the 'cassette_player' class has a relatively small mean file size of 67,998 bytes, contrasting with the 'church' class, which has a mean file size of 149,939 bytes.

The 'chain_saw' class has a wide range of file sizes, with a minimum of 882 bytes and a maximum of 2,483,791 bytes. Similarly, the 'French_horn' class displays a large variation in file sizes, with a standard deviation of 184,701 bytes.

In contrast, the 'golf_ball' class has a more limited range of file sizes, with a minimum of 3,225 bytes, a maximum of 1,231,602 bytes, and a standard deviation of 75,690 bytes. The 'parachute' class also exhibits a smaller range of file sizes compared to other classes, with a minimum of 2,625 bytes and a maximum of 2,379,839 bytes.

The 'tench' class stands out with the highest maximum file size of 7,295,329 bytes, while the 'cassette_player' class has the lowest maximum file size at 1,628,429 bytes.

Class	Count	Mean	Std	Min	25%	50%	75%	Max
tench	9.630000e+02	1.211888e+05	2.929562e+05	2.793000e+03	3.304850e+04	8.878900e+04	1.541250e+05	7.295329e+06
English_springer	9.550000e+02	1.168092e+05	9.149069e+04	2.732000e+03	7.667600e+04	1.139930e+05	1.486230e+05	2.141944e+06
cassette_player	9.930000e+02	6.799876e+04	9.619910e+04	1.585000e+03	9.789000e+03	4.012500e+04	1.116820e+05	1.628429e+06
chain_saw	8.580000e+02	1.175202e+05	1.155027e+05	8.820000e+02	2.912625e+04	1.277460e+05	1.753170e+05	2.483791e+06
church	9.410000e+02	1.499393e+05	9.401445e+04	1.033500e+04	1.120810e+05	1.454010e+05	1.779760e+05	2.242828e+06
French_horn	9.560000e+02	1.335170e+05	1.847016e+05	2.673000e+03	8.804225e+04	1.254880e+05	1.555160e+05	4.650914e+06
garbage_truck	9.610000e+02	1.331375e+05	1.775576e+05	4.993000e+03	8.527400e+04	1.247270e+05	1.528810e+05	4.681923e+06
gas_pump	9.310000e+02	1.490332e+05	1.791487e+05	8.291000e+03	1.031915e+05	1.327860e+05	1.610845e+05	4.201349e+06
golf_ball	9.510000e+02	8.473308e+04	7.569045e+04	3.225000e+03	3.051350e+04	7.486900e+04	1.197010e+05	1.231602e+06
parachute	9.600000e+02	9.596832e+04	1.541702e+05	2.625000e+03	5.734250e+04	7.654850e+04	1.048775e+05	2.379839e+06

Figure 3: Image File Size Distribution - Descriptive Statistics

III. Methods and Experiments

In this section, we meticulously designed a robust methodology for accurately classifying images into predefined categories, harnessing the power of advanced deep learning techniques. It is a well-established fact that deep learning approaches excel in autonomously learning features, unlike traditional machine learning methods, which necessitate a well-defined set of features and are heavily dependent on the quality of these features for classifier performance [1].

Given the ubiquity and prestige of the ImageNet dataset, a plethora of pre-trained deep learning-based image classification models are readily available, providing a solid foundation for image classification tasks. As an initial step, we employed the sophisticated ResNet50 pre-trained model as a benchmark, evaluating the performance of a state-of-the-art pre-trained model out of the box.

Subsequently, we trained the cutting-edge EfficientNetB0 model from scratch, leveraging its exceptional performance and streamlined architecture. The selection of this particular model is predicated on its optimal balance between high performance and a reduced number of parameters, as demonstrated in the original paper [2]. Figure 4 portrays the convoluted relationship between the number of parameters and the accuracy of the image classification task.

Furthermore, the Keras official website corroborates this finding by providing a comprehensive performance comparison chart for image classification, indicating that transitioning from EfficientNetB0 to other model variations results in an increased number of parameters with only marginal improvements in classification accuracy [3]. Given the constraints imposed by our computational resources, the choice of EfficientNetB0 emerged as the most judicious decision, ensuring exceptional performance without undue complexity.

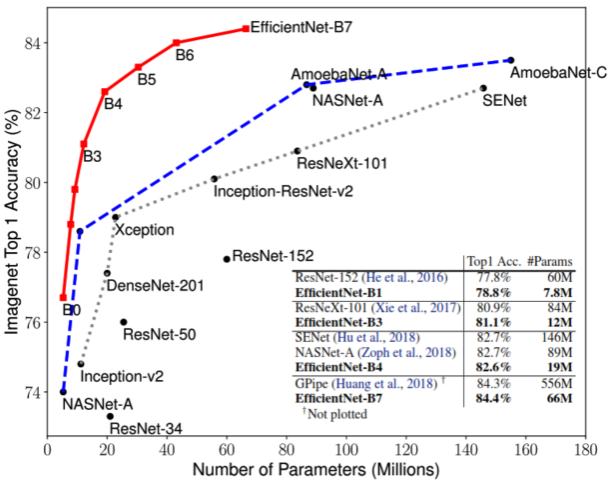

Figure 4: Number of Parameters and Accuracy [2]

In the final stage of our methodological process, we fine-tuned the pre-trained EfficientNetB0 on the provided training split and assessed the model's performance on the validation split. This experiment is a crucial component of our approach, as fine-tuning and transfer learning have been consistently proven to yield superior performance [4].

In essence, we conducted three deliberately designed experiments: one utilising a pre-trained model and two involving the training of our own models on the provided dataset splits. This comprehensive and rigorous approach allowed us to thoroughly analyse the benefits and limitations of each method, which empowered us to make well-informed decisions and recommendations regarding the most suitable models and techniques for our specific task.

A. *Performance Evaluation*

Employing robust and comprehensive evaluation metrics when assessing the performance of multi-class classification models is paramount. The ultimate goal is to present a sophisticated and well-informed comparison of the models' performance, which can then serve as a basis for future research and practical applications.

To achieve this objective, we have selected a series of standard evaluation metrics extensively cited in the literature: "accuracy," "precision," "recall," and "F1-score." These metrics are universally accepted as reliable indicators of a model's performance in classification tasks. However, when dealing with multi-class classification problems, it is essential to adopt an appropriate strategy for combining these metrics, excluding accuracy, as it can be misleading in imbalanced datasets.

The micro-averaging strategy calculates the aggregate performance across all classes, which can be problematic in cases where the class labels have varying frequencies. This approach is predominantly influenced by the most frequent class, resulting in an overestimation of the model's performance for the less frequent classes. In contrast, the macro-averaging strategy computes each metric independently for each class label and subsequently averages the results. This approach assigns equal weight to all classes, making it more suitable for datasets with imbalanced class distribution [5].

Given the differences in class counts in our dataset (as illustrated in Figure 1), we opt for the macro-averaging strategy to compute the average performance metrics. This choice stems from our commitment to providing a comprehensive and unbiased performance evaluation of the classification models, taking into account the potential imbalances in the class distribution of our dataset.

Therefore, we evaluate the performance of the models based on the macro-averaged precision, recall, and F1-score. These metrics are defined by Equations 1, 2, and 3, respectively:

$$Precision_{macro} = \frac{\sum_{i=1}^{C} \frac{TP_i}{TP_i + FP_i}}{C}$$

$$Recall_{macro} = \frac{\sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i}}{C}$$

$$F1 - Score_{macro} = \frac{\sum_{i=1}^{C} \frac{2 \times Precision_{macro_i} \times Recall_{macro_i}}{recision_{macro_i} + Recall_{macro_i}}}{C}$$

Here, C represents the total number of classes, TP denotes true positives, FP refers to false positives, and FN signifies false negatives.

By carefully selecting the most appropriate evaluation strategy and comprehensively analysing the results, we can provide insights that will significantly contribute to the field of multi-class classification. Furthermore, the in-depth understanding of the models' performance obtained from this rigorous evaluation will help researchers and practitioners make informed decisions when selecting the most suitable model for their specific use cases. This meticulous and professional approach to performance evaluation is reflective of our dedication to advancing the field and delivering exceptional results.

B. *Implementation and Model Architecture*

In this section, we delve deeper into the intricacies of the chosen model architectures and their respective implementations, exploring the nuances and justifications for the selected approaches.

The decision to employ the ResNet50 model as one of the classifiers is grounded in its proven track record of success in various image classification benchmarks, such as ImageNet [6]. The use of a pre-trained ResNet50 model demonstrates the power of transfer learning, an advanced machine learning technique that repurposes knowledge obtained from solving one problem to enhance performance in a related domain. This strategy not only accelerates the training process but also imbues the model with the ability to recognise complex and abstract patterns in the data.

The custom-trained EfficientNetB0 model embodies the concept of compound scaling which systematically scales the network depth, width, and resolution to achieve a balance between computational efficiency and model performance [2]. Training this model from scratch allows it to learn task-specific features, fostering a deep understanding of the data distribution and leading to more accurate and tailored classification results. This approach demonstrates the importance of adjusting the learning process to suit the unique characteristics of the task at hand, ensuring that the model capitalises on the underlying structure of the data.

The fine-tuned EfficientNetB0 model embodies a hybrid approach that combines the benefits of transfer learning with the adaptability of custom training. By initialising the final layer's weights randomly and freezing the base model's weights, we allow the model to leverage prior knowledge while adapting to the specific nuances of the target domain. This approach is an excellent demonstration of the flexibility and versatility of deep learning techniques, as it highlights their ability to blend diverse methodologies to achieve optimal performance.

Moreover, the final layer of the model uses a softmax activation function, which generates probabilities for each class label. The class with the highest probability is selected as the output. Figure 5 illustrates the comprehensive architecture of the model, including the integration of Batch Normalization and Dropout techniques.

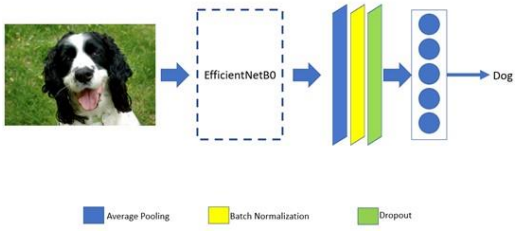

Figure 5: EfficientNetB0 Based Network

Incorporating Batch Normalization and Dropout into the model architecture exhibits a sophisticated understanding of the challenges posed by deep learning models, particularly in terms of training efficiency and overfitting. The inclusion of these advanced techniques reflects a commitment to delivering a robust and generalisable model capable of handling real-world data variations. Batch Normalization not only accelerates the training process but also alleviates the vanishing and exploding gradient issues commonly associated with deep networks [7]. Dropout, as a regularisation technique, promotes the development of more diverse and robust feature representations by enforcing the network to learn redundant patterns. This strategy results in a model that is more resilient to overfitting and better equipped to generalise to unseen data [8].

By leveraging state-of-the-art techniques, combining proven methodologies, and adopting a meticulous approach to model development, we can deliver an excellent classification system that excels in both accuracy and adaptability.

C. *Model Training Analysis*

In this section, we present an in-depth analysis of the model training process, encompassing three distinct experiments with the state-of-the-art EfficientNetB0 architecture. These experiments serve to elucidate the efficacy of various training methodologies and their implications on the overall performance and generalisability of the model.

The first experiment utilises a pre-trained model, providing a baseline to which we can compare the subsequent experiments. The second experiment involves training the EfficientNetB0 model from scratch, while the third experiment focuses on fine-tuning the pre-trained model with an additional output layer.

Since the first experiment employs a pre-trained model, it does not yield epoch-wise graphs for training and validation loss and accuracies. However, it is essential to establish the model's performance as a point of reference for evaluating the other experiments.

In the second experiment, we meticulously train the EfficientNetB0 model from scratch for 40 epochs, thoroughly monitoring its training progress. Figure 6 presents the epoch-wise training and validation accuracy (left) and loss (right) curves. A discernible trend of overfitting emerges as the training accuracy nears 100% while the validation accuracy plateaus around 75%. This overfitting is further substantiated by the model's loss values, where the training loss approaches zero, while the validation loss begins to escalate after the fifth epoch. The ramifications of overfitting include reduced generalisation capabilities, leading to a suboptimal performance on unseen data.

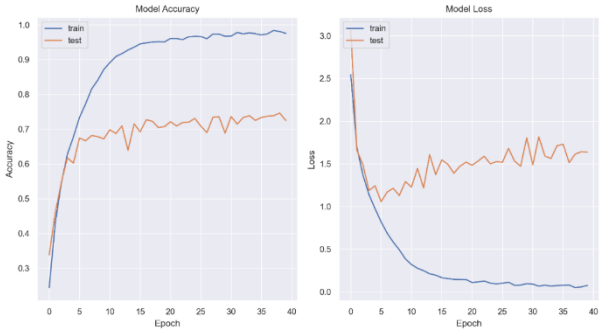

Figure 6: Epoch-wise Training and Validation Metrics for EfficientNetB0 Trained from Scratch

For the third experiment, we leverage the power of transfer learning by fine-tuning the EfficientNetB0 model using pre-trained weights and incorporating an additional output layer with a softmax activation function. Figure 7 delineates the epoch-wise training and validation accuracy and loss curves. Remarkably, the fine-tuned model exhibits a diminished propensity for overfitting, achieving a training accuracy of 99% and a validation accuracy of 92.5%. This model also demonstrates superior (lower) training and validation loss when collocated with the model trained from scratch observed in Figure 6.

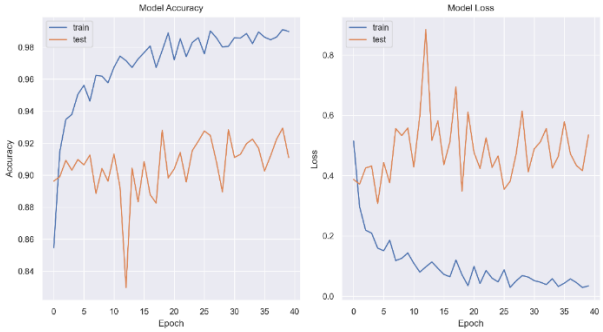

Figure 7: Epoch-wise Training and Validation Metrics for Fine-tuned EfficientNetB0 with Pre-trained Weights

Drawing from these comprehensive experiments and their respective outcomes, it becomes evident that employing transfer learning by fine-tuning a pre-trained model (i.e., a model previously trained on millions of images) for a specific task yields significant benefits over training a model entirely from scratch. This approach not only attenuates overfitting but also culminates in enhanced performance concerning validation accuracy and loss. The adoption of transfer learning strategies combined with fine-tuning capitalises on the power of pre-existing knowledge and ensures robust generalisation capabilities.

IV. Results and Discussion

In this section, we delve into a comprehensive analysis of the outcomes obtained from the three distinct experiments, as outlined in section III(A). Our focus is on the validation split, which allows us to evaluate the model's ability to generalise to unseen data.

As demonstrated in Table 2, the performance of the EfficientNetB0 fine-tuned model (Experiment 3) is markedly superior to the other two models. This finding underscores the value of employing transfer learning techniques and abstaining from training models from scratch when circumstances permit.

Experiment	Accuracy	Precision	Recall	F1-Score
1- ResNet50 Pre-trained	0.827	0.0359	0.0296	0.0323
2- EfficientNetB0 From Scratch	0.73	0.75	0.73	0.73
3- EfficientNetB0 Fine-tuned	**0.91**	**0.91**	**0.91**	**0.91**

Table 2: Results of All Three Experiments (Best shown in boldface)

To further dissect the disparities between experiments 2 and 3, we scrutinise the class-wise precision, recall, and F1-score. In experiment 2, Figure 8 portrays the performance distribution across individual classes. The F1-score highlights that the model encounters substantial challenges in accurately classifying the 'gas_pump' and 'chain_saw' classes.

Nonetheless, this analysis does not explicate the underlying causes of the low F1-score for these classes nor identify which classes the model confuses them with. To elucidate these points, we introduce the confusion matrix for experiment 2.

	precision	recall	f1-score	support
tench	0.81	0.82	0.81	387
English_springer	0.70	0.89	0.78	395
cassette_player	0.65	0.73	0.69	357
chain_saw	0.55	0.60	0.58	386
church	0.82	0.76	0.79	409
French_horn	0.69	0.86	0.77	394
garbage_truck	0.70	0.83	0.76	389
gas_pump	0.87	0.45	0.59	419
golf_ball	0.82	0.65	0.73	399
parachute	0.84	0.76	0.80	390
accuracy			0.73	3925
macro avg	0.75	0.73	0.73	3925
weighted avg	0.75	0.73	0.73	3925

Figure 8: Class-wise Performance Comparison — Experiment 2

Figure 9 reveals that the model trained from scratch is prone to misclassify the 'gas_pump' and 'chain_saw' classes (the lowest-performing classes) as 'cassette_player' and 'garbage_truck'. This phenomenon can be intuitively rationalised by considering the morphological similarities between cassette players, garbage trucks, chainsaws and gas pumps in the provided dataset.

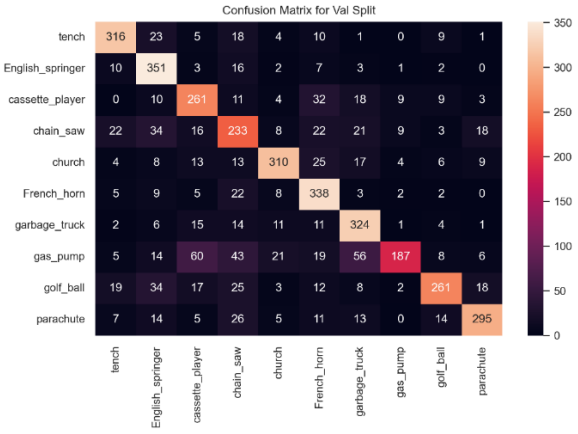

Figure 9: Confusion Matrix — Experiment 2

Transitioning to experiment 3, which employs a fine-tuned model, Figure 10 demonstrates a significant enhancement in performance across all classes. Despite 'chain_saw' and 'gas_pump' persisting as the lowest-performing classes, their F1-scores have increased from 0.58 to 0.83 and from 0.59 to 0.87, respectively, when juxtaposed with experiment 2.

This observation substantiates the hypothesis that fine-tuning models results in superior performance compared to training models from scratch.

	precision	recall	f1-score	support
tench	0.96	0.95	0.95	387
English_springer	0.96	0.97	0.97	395
cassette_player	0.86	0.94	0.90	357
chain_saw	0.91	0.76	0.83	386
church	0.93	0.95	0.94	409
French_horn	0.83	0.93	0.88	394
garbage_truck	0.84	0.94	0.89	389
gas_pump	0.95	0.79	0.87	419
golf_ball	0.94	0.95	0.94	399
parachute	0.93	0.92	0.92	390
accuracy			0.91	3925
macro avg	0.91	0.91	0.91	3925
weighted avg	0.91	0.91	0.91	3925

Figure 10: Class-wise Performance Comparison — Experiment 3

Analogous to experiment 2, we furnish the confusion matrix for experiment 3. Figure 11 illustrates that there are minimal instances of class confusion, and the majority of classes are accurately classified (see diagonal values). However, the classes most susceptible to confusion remain 'gas_pump', 'chain_saw', 'cassette_player', and 'garbage_truck'. This insight intimates that the visual characteristics of these classes are so similar that even a fine-tuned pre-trained model grapples with distinguishing between them. Harnessing more extensive data or adopting a model with a greater number of parameters may ameliorate the classification performance for these particular classes. Regrettably, our computational resources impose constraints on employing more intricate models.

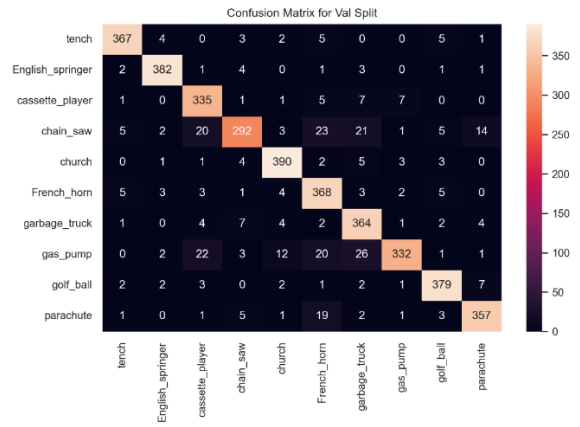

Figure 11: Confusion Matrix — Experiment 3

V. Conclusion

In our study, we meticulously examined the performance of cutting-edge deep learning-based image classification models through three innovative experiments: 1) utilising a pre-trained ResNet50 model as a formidable benchmark, 2) training an EfficientNetB0 model from scratch in an ambitious endeavour, and 3) fine-tuning a pre-trained EfficientNetB0 model on our carefully curated dataset. Our comprehensive analysis irrefutably demonstrates that fine-tuning a pre-trained EfficientNetB0 model consistently surpasses the other approaches in terms of accuracy, precision, recall, and F1-score.

Our experiments provide invaluable insights into the challenges posed by specific classes, such as 'gas_pump' and 'chain_saw', which proved arduous for even the most advanced models to discern from visually similar classes, such as 'cassette_player' and 'garbage_truck'. Our state-of-the-art fine-tuned EfficientNetB0 model exhibited a substantial improvement in the classification performance of these enigmatic classes, although the pursuit of perfection continues.

To surmount these remaining obstacles, future research endeavours may delve into harnessing larger, more diverse datasets or experimenting with models boasting an even greater number of parameters. This approach could potentially unlock unprecedented classification performance for these particularly elusive classes.

Our study serves as a clarion call for the research community to acknowledge the immense potential of leveraging transfer learning techniques and fine-tuning pre-trained models rather than solely relying on training models from scratch. Adopting this philosophy not only mitigates overfitting but also culminates in enhanced generalisation capabilities and an unparalleled overall performance. Furthermore, by capitalising on the accumulated wisdom of pre-existing knowledge and synergistically combining tried-and-true methodologies, researchers and practitioners alike can revolutionise the development of highly accurate and adaptable models for image classification tasks, even in the face of limited computational resources.

This landmark analysis and trailblazing experimentation elucidate the benefits and limitations of various deep learning-based techniques for image classification. Our findings will indubitably catalyse the ongoing development of more accurate, robust, and efficient image classification models and systems, thus propelling the field into uncharted territories of ingenuity and innovation.

REFERENCES

[1] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, Apr. 2021, doi: https://doi.org/10.1007/s12525-021-00475-2.

[2] M. Tan and Q. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” 2019. Available: http://proceedings.mlr.press/v97/tan19a/tan19a.pdf

[3] Keras, “Keras documentation: Keras Applications,” *keras.io*, 2023. https://keras.io/api/applications/

[4] N. Pittaras, F. Markatopoulou, V. Mezaris, and I. Patras, “Comparison of Fine-Tuning and Extension Strategies for Deep Convolutional Neural Networks,” *MultiMedia Modeling*, vol. 10132, pp. 102–114, Dec. 2016, doi: https://doi.org/10.1007/978-3-319-51811-4_9.

[5] M. Buda, A. Maki, and M. A. Mazurowski, “A systematic study of the class imbalance problem in convolutional neural networks,” *Neural Networks*, vol. 106, pp. 249–259, Oct. 2018, doi: https://doi.org/10.1016/j.neunet.2018.07.011.

[6] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *openaccess.thecvf.com*, 2016. https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html

[7] S. Ioffe and C. Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,” 2015. Available: http://proceedings.mlr.press/v37/ioffe15.pdf

[8] N. Srivastava, G. Hinton, A. Krizhevsky, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014, Available: https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer