# ENEL 645 Final Project: Image Classification of Food Dataset


Aemen Mohsin
*Dept. of Electrical and Computer Engineering*
aemen.mohsin@ucalgary.ca

Muhammad Muiz
*Dept. of Electrical and Computer Engineering*
muiz.muhammad1@ucalgary.ca

Soumini Mohandas
*Dept. of Electrical and Computer Engineering*
soumini.mohandas@ucalgary.ca

Tyler Shenassa
*Dept. of Electrical and Computer Engineering*
tyler.shenassa@ucalgary.ca



***Abstract*—Food image classification isn't just about recognizing what's on the plate; it serves as a cornerstone in computer vision with far reaching implications ranging from dietary monitoring and assessment to restaurant recommendation systems. This study focuses on a comprehensive analysis of deep learning models specifically designed for image classification using the Food-101 dataset. Comprising 101 categories and exhibiting substantial variation, this dataset encapsulates complexities with a myriad of noise factors and varying image qualities. Our investigation and analysis involve the use of popular convolutional neural network architectures, such as ResNets and EfficientNets, coupled with various training configurations with respect to learning rates, schedulers, batch sizes, and augmentations. Our objective is to identify the optimal model architecture and training strategy to achieve the highest accuracy in classifications, thereby enabling further advancements in domains such as healthcare, nutrition, and the food industry.**



***Index Terms*—ResNet, EfficientNet, Image Classification, Food Recognition**


## I. Introduction

The advancement of technology in recent years has led the field of computer science, specifically computer vision techniques aimed at interpreting and understanding visual content to achieve significant advancements. One such application of computer vision is image classification for foods, which holds a large potential and opportunity for dietary assessment, health monitoring, and culinary recommendation systems [1]. However, the accurate classification of food images presents unique challenges on a model-to-model basis due to the wide volatility in image properties such as food appearance, lighting conditions, colors, and image quality.

The Food-101 dataset helps to address these challenges by serving as a benchmark for evaluating the effectiveness and performance of various deep learning models in food image classification. This dataset comprises 101 food categories, each consisting of 750 training images and 250 validation images, totalling 101,000 images. To add a layer of complexity, the training set contains intentional noise while the test images have been manually reviewed and cleaned of any noise [2].

In this study, we investigate the performance of various deep learning models on the Food-101 dataset. Specifically, we explore the effectiveness of various architectures, which have demonstrated remarkable performance in image classification, such as ResNets and EfficientNets. Additionally, we explore various training strategies such as adjusting batch sizes, switching learning rates, schedulers, along with various experiments regarding augmentation techniques to enhance model generalization and robustness.

The goal is to identify the most effective approach for accurate food image classification through systematically evaluating different ResNet and EfficientNet models and training configurations. Insights gained from this study can open doors and lead to potential contributions towards the development of more efficient and reliable systems for food recognition, specifically dietary analysis with a focus on implications for healthcare, nutrition, and food industry applications.

## II. Related Work

Within the sphere of deep learning for food image recognition, the efficacy of DCNNs and transfer learning has been highlighted through several noteworthy studies. [3] illustrates how EfficientNet and ResNet models were deployed on the Food-101 dataset, addressing the nuanced challenges of classifying similar food categories and the transformative impact of deep learning on recognition capabilities. [4] delves into a comparative analysis using models like ResNet-50 and ResNext-50 on the UEC Food-100 dataset, identifying ResNext-50's superior accuracy as evidence of certain architectures' advantages in this domain. [5] outlines an approach utilizing a ResNet-50 based DCNN, fine-tuned on multiple datasets including ETHZ-FOOD101 and UECFOOD256, to demonstrate the practical application of transfer learning in optimizing computational resources. Our study, which applies EfficientNet and ResNet models to the Food-101 dataset, both draws inspiration from and contributes to this body of research, showcasing the ongoing advancements in leveraging complex neural network architectures and transfer learning for enhanced food image classification accuracy.

## III. Materials and Methods

### A. Dataset

The Food-101 dataset was utilized for this experiment. The dataset contains 101 classifications, 750 training and 250 validation images for each classification. The dataset holds intentional noise for training images and has been manually cleaned for validation images.

### B. Models

The following convolutional neural network (CNN) models were trained and analyzed for image classification:

- ResNet-18
- ResNet-34
- ResNet-50
- ResNet-101
- EfficientNet-B4
- EfficientNet-B5
- EfficientNet-B6
- EfficientNet-B7

### C. Training Configuration

1) *Iterations and Epochs*
   a) Each model was trained for multiple iterations, with each iteration comprising 15 epochs. Some models ran 10 and 20 epochs.
   b) Iterations for each model were changed on a case-by-case basis due to time constraint.
2) *Learning Rate (LR)*
   a) Different Learning rates were tested for each model and adjusted accordingly based on performance.
   b) The base LR was 0.0001 but iterations also involved switching to 0.01 and 0.001.
3) *Scheduler*
   a) The base scheduler was StepLR with some iterations deviating to ExponentialLr and ReduceOnPlateau.
   b) Learning Step usually involved weight decay of 1e-5 with a step size of 3-5.
4) *Batch Size*
   a) The official batch size for our models was 32 with some deviation to lower numbers due to memory related issues.
5) *Augmentations*
   a) Crop
   b) Rescale
   c) Horizontal flip
   d) Vertical flip
   e) Random rotation
   f) Normalize
   g) Tensor

### D. Evaluation Metrics

The recorded metrics for the performance evaluation of each model involved training and validation accuracies. These were the primary metrics in determining whether a model was underfitting, overfitting, or a good fit.

### E. Experimental Procedure and Methodology

The investigation involved an array of systematic adjustments to the learning rate, weight decay, and the number of training epochs within both ResNet and EfficientNet models. For the models, the learning rate was tested at 0.0001, 0.001, and 0.01 to discern its effect on the learning capability of the models. The weight decay was varied across 1e-5, 1e-4, and 1e-3 to understand its regularization impact on the models' ability to generalize. Training epochs were set primarily at 15, with additional assessments at 10 and 20 epochs to determine the effect of training duration. The optimization algorithms implemented in this study were Adam, AdamW, and RMSprop, each selected to provide insights into the most efficacious optimization strategy for deep learning in food recognition.

## IV. Results and Discussion

### A. ResNet-18

When analyzing the results for both ResNet-18 and ResNet-34 it was evident that neither model was a good fit to the dataset. In both instances, the models achieved lower performance scores on the training data compared to the validation data, suggesting that both models were underfitting [6]. ResNet-18 provided lower scores compared to ResNet-34, although they were both relatively in the same scoring range. The training scores were in the 0.49 to 0.51 range, while the validation scores were all around 0.56, as seen in Fig. 1. The main differences in the iterations were the change in the scheduler to attempt to better the score. The schedulers used were ExponentialLR, StepLR, and ReduceOnPlateau as displayed in Table I. Out of these three schedulers, the ReduceOnPlateau scheduler performed the best. This might be due to the slightly increased learning rate of 0.001, compared to the learning rate of 0.0001, which was used for the other two iterations. ReduceOnPlateau might have also performed slightly better, since after a certain number of epochs, the validation loss started to plateau, which this scheduler was designed to overcome by reducing the learning rate [7].

### B. ResNet-34

For the ResNet-34 model, the same schedulers were used, but the results differed slightly. The general range of the training results was between 0.51 to 0.52, as displayed in Fig. 1, while validation was all 0.58. In this model, the best scheduler was the ExponentialLR scheduler as viewed in Table II. This scheduler had the least disparity between the training and validation scores; however, the training score was still lower. For both models, 15 epochs were used with a batch size of 32. The augmentations used were consistent, with cropping and resizing the images, flipping them horizontally and vertically, normalizing, and converting them into tensors. These augmentations increased the size of the dataset for the training data [8]. As mentioned before, ResNet-34 yielded better scores, but the model was still underfitting the data as depicted in Fig. 3. Due to this, ResNet-50 and ResNet-101 were used for a better fit.

### C. ResNet-50 and ResNet-101

The results from the experiments underscored how adjusting hyperparameters can significantly impact the ResNet-50 and ResNet-101 models' accuracies. The data revealed a disparity of approximately 0.07-0.10 between training and validation accuracies, indicating some degree of overfitting as illustrated in Table III and IV. The most favorable outcome for the ResNet-50 model was observed in iteration 5 where the training accuracy reached 0.9064 and the corresponding validation accuracy was 0.8291 as depicted in Fig. 1, translating to a relatively modest overfitting margin of 0.0773 as seen in Fig. 3. This was achieved under the conditions of a 0.0001 learning rate, a weight decay set at 1e-3, and a training duration of 15 epochs with the Adam optimizer. Meanwhile, the ResNet-101 model's optimal performance was recorded at iteration 6 with a training accuracy of 0.9285 and a validation accuracy of 0.8418, resulting in a slightly higher but still minimal overfitting percentage of 0.0867. This performance was obtained with an identical learning rate and weight decay as the resNet-50, but with a reduced number of epochs set at 10, also using the Adam optimizer. A higher learning rate substantially diminished accuracy across both models, suggesting that lower learning rates are more conducive to this classification task. Alterations to the weight decay and number of epochs further influenced the models' accuracies, highlighting the need for nuanced tuning of these parameters. When comparing optimization algorithms, Adam emerged as a consistent and robust choice, although AdamW and RMSprop provided competitive results, warranting additional exploration.

### D. EfficientNet-B4

EfficientNet-B4 experimentation introduced the preparation and analysis of various augmentations, primarily focusing on resizing the input images [9]. The main configurations for this model involved using a learning rate of 0.0001, Adam optimizer with a weight decay of 1E-5, and ran for 15 epochs on each iteration using a batch size of 32. Each iteration applied consistent augmentations to the input images, including horizontal flip, vertical flip, random rotation of 30 degrees, and resizing. Notably, input resolutions of 224 x 224, 300 x 300, and 380 x 380 were tested across iterations. The first iteration yielded a training and validation score of 0.9173 and 0.8184. The second iteration yielded a training and validation score of 0.9405 and 0.8503. The most favorable outcome for this model was observed in iteration 3 where the training accuracy reached 0. 0.9502 and the corresponding validation accuracy was 0.8691 as depicted in Fig. 2, translating to a relatively modest overfitting margin of 0.0811 as seen in Fig. 4. Table V reveals a decrease in overfitting alongside enhanced training and validation scores, suggesting improved generalizability as the models approach recommended input image resolutions.

### E. EfficientNet-B5

EfficientNet-B5 testing primarily focused on adjustment of weight decay in order to reduce overfitting [10]. Other parameters were kept consistent in order to see the direct influence of changing weight decay on accuracy and generalization. A learning rate of 0.0001, 15 epochs, and a batch size of 32 were used. Augmentation included an input resolution of 224x224, horizontal flip, vertical flip, and random rotation (up to a certain degree). Weight decays of 1E-5, 1E-4, and 1E-3 were tested. This resulted in training / validation scores of 0.9231 / 0.8186, 0.9242 / 0.8220, and 0.8956 / 0.8164, respectively, as shown in Table VI. These results suggest that while using EfficientNet-B5, solely changing weight decay does not have significant influence on overfitting as the lowest overfitting value was 0.0792 in iteration 3 as seen in Fig. 4. It is likely that other parameters, such as dropout, should be changed in conjunction with weight decay to reduce overfitting [11].

### F. EfficientNet-B6

EfficientNet-B6 Used similar configurations to the test involving EfficientNet-B4. However, the increased model complexity demanded better hardware, limiting the extent of resizing and augmentation experimentation. Due to the limited specifications available, resizing was suboptimal. Specifically, for the first iteration, a batch size of 32 and an input resolution of 224 x 224 were used. The second iteration used a batch size of 16 and an input resolution of 300 x 300, while the last iteration used a batch size of 32 and an input resolution of 300 x 300. The corresponding training and validation scores for the first and second iterations were 0.9241/0.8226 and 0.9129/0.8590 respectively. Lastly, the third iteration yielded a training and validation score of 0.9305/0.8448 as seen in Table VII. Although not all scores increased, the overall overfitting percentage of each iteration decreased, with the exception of iteration 3. This results in iteration 2 being the best model, with the least overfitting value of 0.0539 as seen in Fig. 4. Ideally, based on prior experiments with EfficientNet-B4, EfficientNet-B6 models should utilize an input resolution of 528 x 528. When this is coupled with a smaller batch size of 16, it is anticipated to produce better outcomes, reflecting the balance between computational efficiency and model accuracy.

### G. EfficientNet-B7

EfficientNet-B7 testing faced issues with hardware limitations, due to its increased complexity compared to other models [12]. Iterations focused on adjusting parameters to maximize use of available memory. A learning rate of 0.0001, weight decay of 1E-5, and 15 epochs were used. Augmentation included a horizontal flip, vertical flip, and random rotation. Initial batch size was 16, increasing up to 24. Past this point the model could not be run due to memory limitations. Input resolution was also limited to 224x224 due to memory limitations. Additionally, each epoch using EfficientNet-B7 had 2-4x the runtime of EfficientNet-B4 and B5 models. Despite these constraints, the model achieved consistent training scores of 0.95, and validation scores 0.84 as seen in Table VIII. As seen in Fig. 4, the high overfitting percentage of 0.1107 proves that this model is not optimal.

### H. Figures and Tables

TABLE I
RESNET-18 MODEL RESULTS

| Iter | LR | Scheduler | Epoch | Batch | Augmentation | Train, Val |
|---|---|---|---|---|---|---|
| 1 | 0.0001 | Expo LR | 15 | 32 | Crop, Flip, Tensor, Normal | 0.4989, 0.5636 |
| 2 | 0.0001 | StepLR | 15 | 32 | Crop, Flip, Tensor, Normal | 0.4975, 0.5604 |
| 3 | 0.001 | Reduce On Plateau | 15 | 32 | Crop, Flip, Tensor, Normal | 0.5109, 0.5657 |

TABLE II
RESNET-34 MODEL RESULTS

| Iter | LR | Scheduler | Epoch | Batch | Augmentation | Train, Val |
|---|---|---|---|---|---|---|
| 1 | 0.0001 | Expo LR | 15 | 32 | Crop, Flip, Tensor, Normal | 0.5218, 0.5895 |
| 2 | 0.0001 | StepLR | 15 | 32 | Crop, Flip, Tensor, Normal | 0.5161, 0.5853 |
| 3 | 0.0001 | Reduce On Plateau | 15 | 32 | Crop, Flip, Tensor, Normal | 0.5154, 0.5827 |

TABLE III
RESNET-50 MODEL RESULTS

| Iter | LR | LR Step | Epoch | Batch | Optimizer | Train, Val |
|---|---|---|---|---|---|---|
| 1 | 0.0001 | 1E-5 | 15 | 32 | Adam | 0.9161, 0.8307 |
| 2 | 0.001 | 1E-5 | 15 | 32 | Adam | 0.8644, 0.7804 |
| 3 | 0.01 | 1E-5 | 15 | 32 | Adam | 0.3001, 0.2893 |
| 4 | 0.0001 | 1E-4 | 15 | 32 | Adam | 0.9136, 0.8273 |
| 5 | 0.0001 | 1E-3 | 15 | 32 | Adam | 0.9064, 0.8291 |
| 6 | 0.0001 | 1E-3 | 10 | 32 | Adam | 0.9080, 0.8279 |
| 7 | 0.0001 | 1E-3 | 20 | 32 | Adam | 0.9061, 0.8275 |
| 8 | 0.0001 | 1E-3 | 15 | 32 | AdamW | 0.9162, 0.8263 |
| 9 | 0.0001 | 1E-3 | 15 | 32 | RMSprop | 0.9069, 0.8279 |

TABLE IV
RESNET-101 MODEL RESULTS

| Iter | LR | LR Step | Epoch | Batch | Optimizer | Train, Val |
|---|---|---|---|---|---|---|
| 1 | 0.0001 | 1E-5 | 15 | 32 | Adam | 0.9426, 0.8400 |
| 2 | 0.001 | 1E-5 | 15 | 32 | Adam | 0.8755, 0.7887 |
| 3 | 0.01 | 1E-5 | 15 | 32 | Adam | 0.2459, 0.2354 |
| 4 | 0.0001 | 1E-4 | 15 | 32 | Adam | 0.9417, 0.8382 |
| 5 | 0.0001 | 1E-3 | 15 | 32 | Adam | 0.9331, 0.8402 |
| 6 | 0.0001 | 1E-3 | 10 | 32 | Adam | 0.9285, 0.8418 |
| 7 | 0.0001 | 1E-3 | 20 | 32 | Adam | 0.9331, 0.8400 |
| 8 | 0.0001 | 1E-3 | 10 | 32 | AdamW | 0.9414, 0.8358 |
| 9 | 0.0001 | 1E-3 | 10 | 32 | RMSprop | 0.9289, 0.8398 |

TABLE V
EFFICIENTNET-B4 MODEL RESULTS

| Iter | LR | LR Step | Epoch | Batch | Augmentation | Train, Val |
|---|---|---|---|---|---|---|
| 1 | 0.0001 | 1E-5 | 15 | 32 | 224x244, Flip, Rotation | 0.9173, 0.8184 |
| 2 | 0.0001 | 1E-5 | 15 | 32 | 300x300, Flip, Rotation | 0.9405, 0.8503 |
| 3 | 0.0001 | 1E-5 | 15 | 32 | 380x380, Flip, Rotation | 0.9502, 0.8691 |

TABLE VI
EFFICIENTNET-B5 MODEL RESULTS

| Iter | LR | LR Step | Epoch | Batch | Augmentation | Train, Val |
|---|---|---|---|---|---|---|
| 1 | 0.0001 | 1E-5 | 15 | 32 | 224x244, Flip, Rotation | 0.9231, 0.8186 |
| 2 | 0.0001 | 1E-4 | 15 | 32 | 224x244, Flip, Rotation | 0.9242, 0.8220 |
| 3 | 0.0001 | 1E-3 | 15 | 32 | 224x244, Flip, Rotation | 0.8956, 0.8164 |

TABLE VII
EFFICIENTNET-B6 MODEL RESULTS

| Iter | LR | LR Step | Epoch | Batch | Augmentation | Train, Val |
|---|---|---|---|---|---|---|
| 1 | 0.0001 | 1E-5 | 15 | 32 | 224x244, Flip, Rotation | 0.9241, 0.8226 |
| 2 | 0.0001 | 1E-4 | 15 | 16 | 300x300, Flip, Rotation | 0.9129, 0.8590 |
| 3 | 0.0001 | 1E-3 | 15 | 32 | 300x300, Flip, Rotation | 0.9305, 0.8448 |

TABLE VIII
EFFICIENTNET-B7 MODEL RESULTS

| Iter | LR | LR Step | Epoch | Batch | Augmentation | Train, Val |
|---|---|---|---|---|---|---|
| 1 | 0.0001 | 1E-5 | 15 | 16 | 224x244, Flip, Rotation | 0.9497, 0.8390 |
| 2 | 0.0001 | 1E-4 | 15 | 20 | 224x224, Flip, Rotation | 0.9507, 0.8368 |
| 3 | 0.0001 | 1E-3 | 15 | 24 | 224x224, Flip, Rotation | 0.9527, 0.8366 |

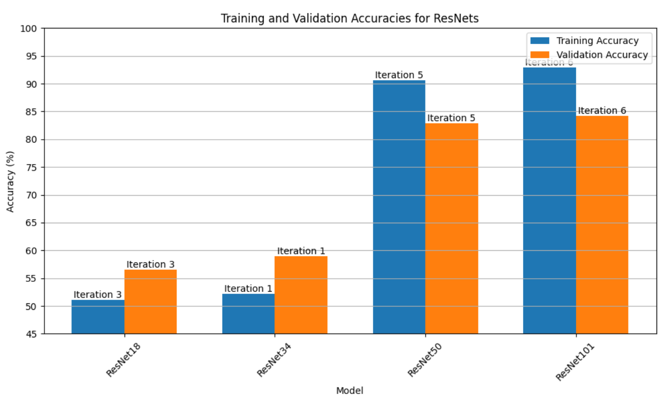


Fig. 1. Best Training and Validation Accuracy for ResNet Models.

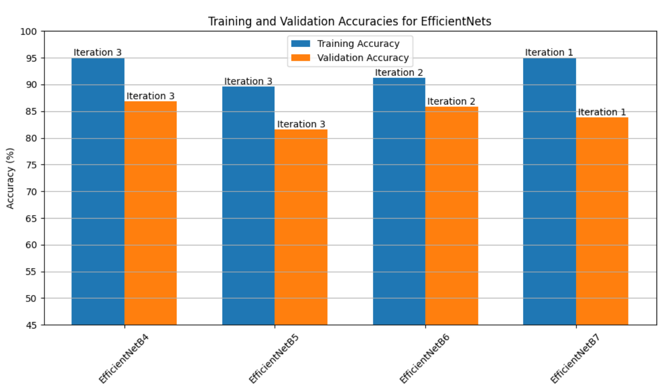


Fig. 2. Best Training and Validation Accuracy for EfficientNet Models.

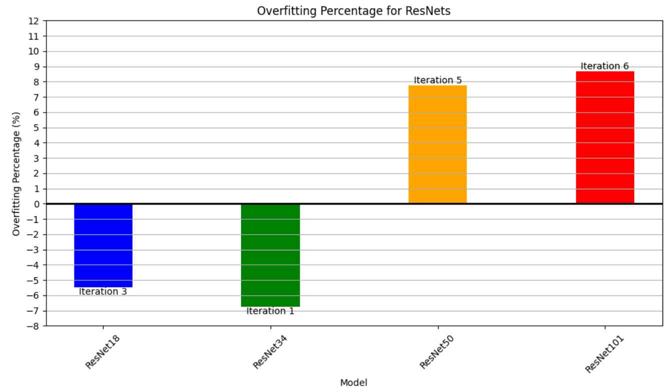


Fig. 3. Overfitting Percentage for ResNet Models.

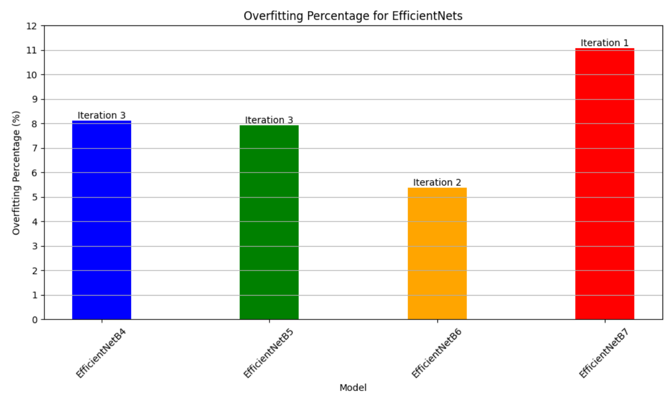


Fig. 4. Overfitting Percentage for EfficientNet Models.

## V. CONCLUSION

Different models were used to train and test the Food-101 dataset, including ResNet-18, ResNet-34, ResNet-50, ResNet-101, EfficientNet-B4, EfficientNet-B5, EfficientNet-B6, and EfficientNet-B7. These models were all fine-tuned using different hyperparameter tuning techniques. After further analysis of the results, it was determined that the second iteration of EfficientNet-B6 was the best model with a training score of 0.9129 and a validation score of 0.8590 since it produced the smallest overfitting value of 0.0539 as seen in Fig. 4. The main limitation of the EfficientNet-B6 model was its demand on computational resources, as the model required an excessively long training period. Moreover, memory limitations were encountered when training this model. Overall, the model's strengths, such as high training and validation scores and minimal overfitting, outweighed its limitations.

# REFERENCES


[1] K. Yanai, and Y. Kawano, "Food image recognition using deep convolutional network with pre -training and fine-tuning," in IEEE International Conference on Multimedia & Expo Workshops, 2015, pp.1-6

[2] L. Bossard, "Papers with code - food-101 dataset," Dataset — Papers With Code, https://paperswithcode.com/dataset/food-101 (accessed Apr. 7, 2024).

[3] Liu, Yizhe. (2023). Automatic food recognition based on efficientnet and ResNet. Journal of Physics: Conference Series. 2646. 012037. 10.1088/1742-6596/2646/1/012037.

[4] S. Memiş, B. Arslan, O. Z. Batur and E. B. Sönmez, "A Comparative Study of Deep Learning Methods on Food Classification Problem," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 2020, pp. 1-4, doi: 10.1109/ASYU50717.2020.9259904.

[5] Zahisham, Zharfan & Lee, Chin-Poo & Lim, Kian. (2020). Food Recognition with ResNet-50. 1-5. 10.1109/IICAIET49801.2020.9257825.

[6] A. Soleymani, "Your validation loss is lower than your training loss? this is why!," Medium, https://towardsdatascience.com/what-your-validation-loss-is-lower-than-your-training-loss-this-is-why-5e92e0b1747e (accessed Apr. 7, 2024).

[7] "Reducelronplateau" ReduceLROnPlateau - PyTorch 2.2 documentation, https://pytorch.org/docs/stable/generated/torch.optim.lr_scheduler.ReduceLROnPlateau.html (accessed Apr. 7, 2024).

[8] Y. Chauhan, "Increase the performance of Image Classification Models:," Medium, https://medium.com/international-school-of-ai-data-science/increase-the-performance-of-image-classification-models-b466e1ae3101 (accessed Apr. 7, 2024).

[9] Yi, San-Li & Yang, Xue-Lian & Wang, Tian-Wei & She, Fu-Rong & Xiong, Xin & He, Jian-Feng. (2021). Diabetic Retinopathy Diagnosis Based on RA-EfficientNet. Applied Sciences. 11. 11035. 10.3390/app112211035.

[10] D. Vasani, "This thing called weight decay," Medium, https://towardsdatascience.com/this-thing-called-weight-decay-a7cd4bcfccab (accessed Apr. 7, 2024).

[11] D. C.-E. Lin, "8 Simple Techniques to Prevent Overfitting," Medium, https://towardsdatascience.com/8-simple-techniques-to-prevent-overfitting-4d443da2ef7d (accessed Apr. 7, 2024).

[12] A. Arasanipalai, "How Efficient is EfficientNet?," Weights & Biases, https://wandb.ai/ajayuppili/efficientnet/reports/How-Efficient-Is-EfficientNet—Vmlldzo4NTk5MQ (accessed Apr. 7, 2024).