Object Recognition Improvements Obtained Through Saliency-Based Image Enhancement

MSc Research Project

Msc Data Analytics

Prachi

Student ID: 21178071

School of Computing
National College of Ireland

Supervisor: Teerath Kumar

National College of
IrelandProject
Submission Sheet
School of
Computing

Student Name:	Prachi
Student ID:	21178071
Programme:	Msc Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Teerath Kumar
Submission Due Date:	14/08/2023
Project Title:	Object Recognition Improvements Obtained Through Saliency-Based Image Enhancement
Word Count:	7,067
Page Count:	30

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at therear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Prachi
Date:	14th August 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator office must be placedinto the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	

Object Recognition Improvements Obtained Through Saliency-Based Image Enhancement

Prachi
21178071

Abstract

Novel methods have driven image processing and computer vision advances. Artificial intelligence and machine learning—inextricably linked—are advancing rapidly. This article discusses saliency augmentation, a game-changing approach that highlights an image's most significant aspects. This satisfies the pressing need for higher performance and interpretability while also encouraging accurate model predictions. Based on the principle of saliency mapping, which mimics the way in which human eyes focus their attention and identifies sections of a picture that demand close investigation, saliency augmentation was developed. Instead of just adding more data to an existing dataset, as is done with typical augmentation methods, saliency augmentation seeks to emphasise significant characteristics while simultaneously eliminating unnecessary noise. Extensive trials on the CIFAR-10 dataset utilising a wide range of pre-trained network designs, including VGGNet19, ResNet, MobileNet, EfficientNet, and DenseNet, indicate the efficacy of saliency augmentation in our study. On FashionMNSIT dataset ResNet and VGG19, saliency worked with more accuracy than random erasing. In particular, Saliency-based Gradient Augmentation outperforms both Normal Augmentation and Random Erasing by a wide margin of around 2.6% across all types of models. In this paper, we will cover the groundwork for saliency augmentation, from its methodology to its comparative analyses to the ethical concerns we want to raise and the promising new areas we hope to investigate. By bridging the gap between human visual cognition and computing capabilities, Saliency Augmentation creates a new paradigm in the field of picture data enhancement. The consequences of this go well beyond the realm of computer vision and into many other fields. Saliency Augmentation is a method that helps computers mimic human visual perception.

Keywords: Normal Augmentation, Random Erasing, CIFAR10, MobileNet, DenseNet, VGG19, Saliency Gradient, EfficientNet, ResNet

1 Introduction

1.1 What is Data Augmentation?

Data augmentation is a technique in machine learning and deep learning that artificially increases dataset size and diversity by applying transformations. It aims to improve the performance and generalization of machine learning models by exposing them to a wider range of variations and scenarios and not just getting restricted to the Training data. With reference to images, it is defined as a set of various techniques applied to the images to transform or edit them to all sorts of possible forms they can exist in or highlight only the relevant parts of the images that are actually required for the fulfillment of the task.

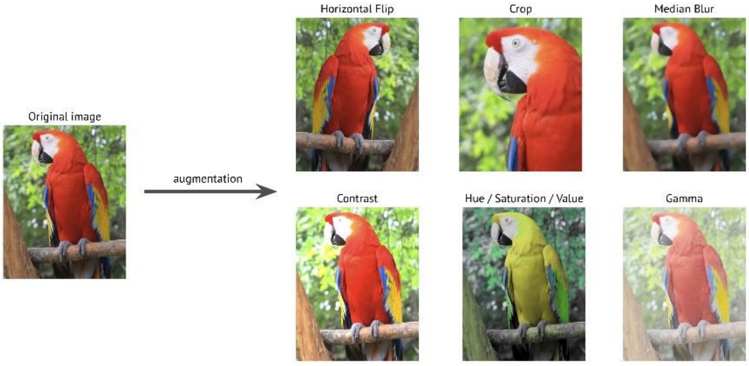


Image input (Source: https://albumentations.ai/)

Figure 1: An example of different functions changing the original

In the above figure we can see that the input image has gone through some changes based on the different functions that has been applied to it.

$$f(i) \ = T(f(i)), where\ (i)\ is\ the\ pixel\ value, T\ is\ the\ Augmenting\ function$$

1.2 What is Saliency Augmentation and what advantages does it have over normal augmentation?

“Saliency Augmentation” is defined as the technique by virtue of which we can highlight those important parts of the images that are required to draw human attention or help in increasing the accuracy of any model while performing a particular task. This process is carried out with the help of an Image Processing/Computer Vision technique called “Saliency Mapping” which imitates human visual attention and determines which parts of an image are most likely to draw human focus or contain some crucial information about the image. Saliency mapping is a useful approach in computer vision that bridges the gap between low-level image processing and high-level visual cognition, allowing computers to prioritize and interpret visual information in a manner similar to human perception.

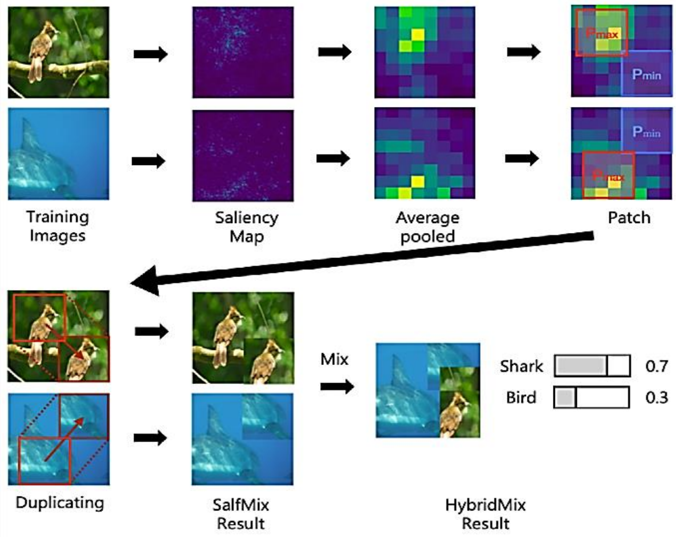


Figure 2: An example depicting the Image Saliency technique with the use of Saliency Gradient (Source: Analytics Vidhya)

1.3 Aim

Rapid progress in areas like artificial intelligence and machine learning has allowed for significant improvements in image processing and computer vision. The aforementioned context is what has allowed for these advancements. The advent of Saliency Augmentation is especially noteworthy here because of its potential impact on the situation. The focus of this technique is on extracting certain regions of an image for the purpose of close examination. Because it provides a significant channel for doing so, it is a crucial instrument for extracting useful knowledge from visual input. For instance, images collected at a crime scene might yield useful information if processed with a technique called saliency enhancement, which could be utilised by law enforcement to solve mysteries. In view of these technical developments and societal concerns, the goals of this study are to

(1) examine the concept of Saliency Augmentation and
(2) conduct a thorough evaluation of pertinent prior studies that have made significant contributions to this specialised field of study.

That is to say, not only will the idea of Saliency Augmentation be investigated, but the term itself will also be explored.

1.4 Motivation

Though there has been some significant research done in this field of study yet a lot needs to be done one of the specific reasons being the unavailability of the most accurate data where only specific and important parts of an image that would truly give us some conclusion are highlighted. Secondly, there have been a lot of restraints by ethics like the privacy of the data, etc. Therefore keeping all those challenges and mind and striving to overcome them in the most efficient way we can, we have presented this paper with some considerations from papers that have made some significant research or impact in this field

1.5 Research Objectives

In this research paper, we would focus on the following objectives-:

1. Deploying the most efficient deep learning or machine learning models for the data augmentation that highlights only the crucial features.
2. Enhancing visual attention based saliency gradient based augmentation technique for detailed object detection classification.

This study begins with a literature review of Saliency Augmentation research, leading to a proposed methodology in Chapter 3. Chapter 4 compares implemented models, highlighting the most effective. Chapter 5 analyzes results, while Chapter 6 concludes findings.

2 Literature Review

To attain the objectives of our research on Saliency Augmentation in the most efficient way we can, we would do a literature review of 24 research papers divided into 3 sections with each section containing a particular concept in which the research has been done and published by many reputed authors who have a lot of expertise in the field of. Some of the research articles which we would use as references for our study are as follows:

2.1 Research done towards Deep Learning and Transfer Learning integrated with Augmentation techniques

For example: (Gorji, S., & Clark, J.J., 2018) A novel method was created by fusing static saliency models with the Attentional Push effect in motion pictures. A multi-stream ConvLSTM network combines saliency and attentional dynamics to improve fixation prediction. This improves fixation prediction by decreasing the entropy between enhanced saliency and viewer gaze patterns.

According to a recent study (C. Shorten & T.M. This review looks at data augmentation in the context of deep learning, where it may be used to address the issue of inadequate data. Meta-learning, extended testing time, and the resolution effect are also explored, along with other methods such geometric modifications and expanded colour spaces. This exemplifies how the upgrade enhances model performance and the possibilities of datasets.

According to a recent study (C. Shorten & T.M. A method for tracking where people's eyes go in social situations uses a concept called shared attention to make that prediction. Fixation prediction accuracy is greatly enhanced by using transfer learning, fine-tuning, and assessment on eye fixation datasets to learn augmented saliency.

Drone multiscale object identification is enhanced by combining models and voting methods in an ensemble transfer learning approach (Walambe, Marathe, & Kotecha, 2021). Improves the basic models' performance and reliability for better object detection in UAV photographs while efficiently addressing the dataset's limitations.

Pre-segmented breast masses in mammograms may be classified as benign or malignant using Convolutional Neural Networks trained using transfer learning and data augmentation (Lévy, D., and Jain, A., 2016). This resolves the issue of incorrect diagnoses and boosts the efficacy of breast cancer screening.

(Ferreira, C. A., & coworkers, 2018) Using Inception Resnet V2 and data augmentation, a deep neural network can classify a breast cancer histological image. achieves 76% accuracy in the blind test set, contributing to breast cancer diagnosis despite little data.

(Fuhl, W., 2022): Combines large recurrent neural networks with transfer learning to make deep neural network designs more effective. makes use of all available layers and even deeper layers to improve model performance at both high and low data augmentation levels.

As reported in (Naushad, R., T. Kaur, & E. E. Ghaderpour, 2021): Classification of remotely sensed images might improve land use and cover mapping through the use of transfer learning. Computational efficiency and accuracy of 99.17% are achieved by combining the Eurostat dataset with early stopping, gradient clipping, adjustable learning rates, and data augmentation to enhance classification performance.

2.2 Research done on different Saliency Augmentation Techniques

This study's authors (An, J., Jang, S., Kwon, J., Jin, K., and Kim, Y., 2022) present a novel method for enhancement by combining patches that do not overlap from prominent areas of an image. This strategy is robust against noise and efficient at exploiting object features. This method of improvement is more resilient to harmful interference than conventional methods. Top-1 accuracy rates of 97.26%, 83.99%, and 82.40% were achieved by Wide ResNet on publicly available datasets, outperforming competing methods.

The method of image contrast enhancement is examined in "Image Contrast Enhancement" (Gu, K. Zhai, Yang, X., Zhang, W., and Chen, C.W., 2014), with a focus on the potential benefits of saliency preservation. This study creates a method for automatically improving contrast in images. It uses an automated parameter picker and a histogram-altering architecture. Aesthetically pleasing and histogram-equalized versions of the original image are also included into the framework. An assessing criterion for visual quality is developed using saliency maintenance as its foundation; this aids in parameter selection. The experiment results show that salience may be kept for a long time, and the effect of amplification is striking.

While deep learning has improved the reliability of computer vision, there are still challenges to be met in order to get the best results with limited amounts of training data. To address these issues, researchers have attempted to include human opinion on crucial visual areas into training data. The technique improves both precision and generality. As a consequence, the technique achieves excellent generalisation in a leave-one-attack-type-out evaluation scenario, and it reduces the mistake produced by the LivDet-Iris 2020 winner. (Czajka, A.; Boyd, A.; Bowyer, K.W.; 2022) To achieve high accuracy and generalisation with less training data, this finding provides a new area of research into the use of human intelligence into deep learning training algorithms.

In order to help readers better understand hidden points of interest, the authors of this piece (Sandor, C., Cunningham, A., Dey, A., & Mattila, V.V., 2010) present an innovative Augmented Reality X-ray visualisation technique. Sandor, Cunningham, Dey, and Mattila are credited with creating the method. The technique makes advantage of colour, brightness, and motion, three more noticeable characteristics. Human testing has shown that the prototype is successful in giving a more complete setting without hindering object selection. However, it does have certain negatives, like a higher degree of adaptation being needed. The results suggest that the suggested representation has the potential to enhance both the user experience and the understanding of hidden details.

Models for identifying visual saliency are used often in vision tasks. These models include things like RGBD and co-saliency and even video. These models take advantage of depth cues, inter-image correlation, and temporal links to identify important regions from RGBD pictures. The purpose of video saliency detection is to pick out objects in moving videos. According to research (Cong, R., Lei, J., Fu, H., Cheng, M.M., Lin, W., and Huang, Q., 2018), This study examines a wide range of saliency detection methods, discusses contemporary issues, and reviews several saliency detection algorithms, as well as providing evaluation datasets and quantitative measures.

To aid in the quick identification and segmentation of core fire zones in aerial photographs, this work (Zhao, Y., Ma, J., Li, X., & Zhang, J., 2018) proposes a unique vulnerability detection approach. The Fire_Net 15-layer self-learning DCNN architecture outperforms previous methods used for real-time inspections of wildfires. It is 98% accurate on average and takes only 41.5 ms to process each image. Forty example photos obtained from news headlines about wildfires were used to assess the effectiveness of the proposed approach. This technique effectively prevents the blurring and loss of detail that results from downsizing in one step.

As an alternative to RGB colour imaging, light field imaging may be used to pinpoint specific regions of interest. CNN-based techniques for saliency identification provide a tough barrier due to their generalizability and insufficient number of accessible datasets for processing light field inputs. For the purpose of training deeper networks and benchmarking, a new Lytro Illum dataset has been released. There are 640 light fields and ground-truth saliency maps available in the collection. The proposed end-to-end CNN-based framework has the potential to be generalised (Zhang, J., Liu, Y., Zhang, S., Poppe, R., and Wang, M., 2020) and outperforms state-of-the-art techniques. It uses three novel MAC blocks to achieve this goal.

In order to complete categorization jobs, run retrieval systems, and archive data, automated description of digital multimedia content is required. This study looks into the challenges of categorising cultural products like the many depictions of Mexican architecture. In order to predict where a viewer's attention would be focused in a picture, a deep convolutional neural network can be trained using a saliency-driven approach. This study examines the impact of image cropping and saliency maps on model behaviour and compares the results to state-of-the-art methods. Testing on a sizable subset of the ImageNet database extends previous work on image padding techniques and generalisation on large-scale generic databases (Obeso, A.M., Benois-Pineau, J., Vázquez, M.G., and Acosta, A.R., 2019; Obeso, Benois-Pineau, and Acosta, 2019).

2.3 Research done toward the integration of variation Saliency Augmentation techniques with Computer Vision

A approach for modifying images is shown in (Mechrez et al., 2019) that makes use of patch-based editing to change object salience while retaining appearance attributes. This allows for better saliency modulation and a more natural appearance than previous methods.

The vast data, advanced processing tools, and unique capture devices that have propelled the collaboration between the computer graphics and computer vision communities are highlighted (Cheng et al., 2017). Potential synergies between the two fields are explored, as are methods for enhancing analysis, manipulation, synthesis, and interaction.

(Ghose et al., 2019) This method incorporates human judgement on important visual areas to solve the challenges with deep learning caused by a lack of training data. Gains in accuracy and generalisation, with fewer errors and impressive transferability in a leave-one-attack scenario, are a step towards incorporating human intelligence into training processes.

A texture-based technique for visual attention, including face recognition signals and local context suppression, is proposed (Kucerova & Sikudova, 2011). As a result, this technique may be used to enhance attention modelling for a wide range of items, including faces.

Saliency detection in 360-degree videos using a sphere-based convolutional neural network (Zhang et al., 2018). Time spent learning is reduced by employing rotation for convolution and parameter sharing for kernels constructed on a spherical crown. Sequential saliency detection employing a sphere-shaped version of the U-Net, in addition to rigorous testing, provides evidence of effectiveness.

Using a center-dark channel prior, (Zhu et al., 2017) introduces a novel approach to identifying important features in RGB-D images. By generating and fusing saliency maps, it outperforms the state-of-the-art methods for identifying visually striking objects.

To facilitate action recognition, Mathe and Sminchisescu (2012) proposed a method that combined human eye motions with dynamic computer vision datasets. Using examples to demonstrate the stability of visual search patterns and to motivate the development of end-to-end trainable computer vision systems based on human fixations, this article presents dynamic consistency and alignment models.

A recurrent attentional convolutional-deconvolution network (RACDNN) is proposed as a saliency detection method (Kuen et al., 2016). Outperforms state-of-the-art algorithms on challenging datasets by iteratively responding to image sub-regions, refining saliency, and learning context-aware features. Furthermore, it is capable of learning traits that are sensitive to context.

2.4 Summary

Research initiatives with the goal of improving various aspects of computer vision through the use of innovative approaches and tactics are represented in the compiled findings. Combining static saliency models with the Attentional Push effect, as demonstrated by Gorji and Clark (2018), improves fixation prediction. One of the more recent works of this kind was this one. Shorten and T.M.'s research on the use of data augmentation in deep learning demonstrates the effectiveness of the method in addressing issues related to a shortage of data and improving model performance generally. Recent studies have shown that transfer learning and fine-tuning are effective approaches to improve prediction accuracy, especially in tasks like eye fixation prediction and land use classification. Both of these approaches had the same result. Drone-based object identification also benefits from ensemble transfer learning's ability to effectively manage data constraints while simultaneously boosting detection performance. It appears that the proper categorization of pre-segmented breast masses in mammograms is greatly aided by the use of Convolutional Neural Networks in combination with transfer learning and data augmentation, all of which aid in the identification of breast cancer. Like Fuhl's approach, this one also employs recurrent neural networks and transfer learning to improve deep neural network architectures. There is a great deal of promise in combining saliency enhancement methods with computer vision, as evidenced by a number of research. In this study, new strategies are developed to increase the prominence of visual content while preserving its aesthetic value. In addition, sphere-based convolutional neural networks and texture-based methods are explored for their application in saliency identification. Overall, these results help advance computer vision by expanding our knowledge of and ability to use saliency augmentation techniques. They also show the approaches' potential to boost accuracy, robustness, and generality across many other visual tasks.

2.5 Research Niche

The goal of the system that has been presented is to conduct an exhaustive study of a variety of augmentation approaches using well-established datasets such as CIFAR-10, while simultaneously presenting an innovative approach to saliency augmentation. Following the training of a variety of deep learning models, including several architectures such as CNNs, the system will apply both conventional augmentations and the innovative saliency-based technique to the dataset in an orchestrated pipeline. This will allow for the training of a wide range of deep learning models. The purpose of the study is to determine the impact that various augmentation procedures, including the unique saliency augmentation, have on model accuracy and generalisation. This will be accomplished by carefully examining the performance of the models making use of conventional metrics such as accuracy. The system will give vital insights into the interplay between augmentation approaches and model behaviour through this iterative process, which will further the knowledge of successful data augmentation strategies in the context of picture classification problems.

3 Methodology

3.1 Research Resource: Cifar10 dataset

Below displays 5 randomly selected CIFAR-10 dataset images in a horizontal row, each with its index in a subplot. The images are shown without axis labels.

Figure 3: CIFAR10 randomly chosen 5 images

3.2 Research Method: Machine Learning Models

3.2.1 DenseNet

Densenet, also referred to as Dense Convolutional Network, represents a significant advancement in deep learning architecture, introduced by Gao Huang and his colleagues in 2016. This innovative design effectively addresses the vanishing gradient issue while fostering feature reuse, ultimately leading to enhanced efficiency and performance. The core principle of Densenet revolves around creating dense connections between layers, where each layer receives input from all preceding layers. This facilitates comprehensive information propagation and knowledge sharing across the network, boosting feature propagation and gradient flow during training, resulting in improved optimization. Notably, Densenet attains remarkable efficiency by minimizing the required number of parameters, making it computationally less intensive and more memory-efficient compared to conventional architectures.

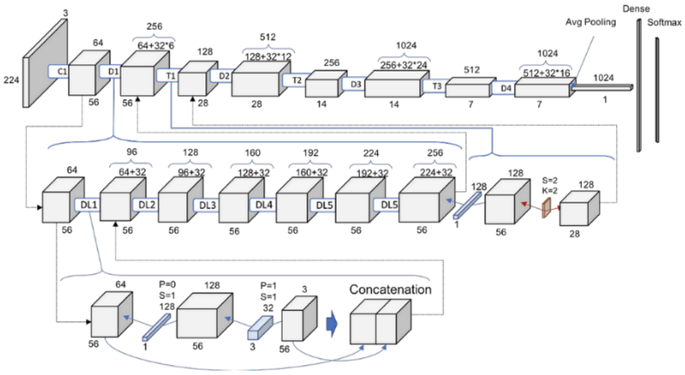


Figure 4: Architecture of DenseNet (Gao Huang. et. al., 2016)

3.2.2 MobileNet

MobileNet, introduced by Andrew G. Howard and his team in 2017, stands as a pioneering deep learning architecture tailored for efficient image processing and classification on resource-constrained devices. MobileNet strategically utilizes depth-wise separable convolutions to significantly lower the computational complexity inherent in conventional convolutions. This approach hinges on factorizing a standard 3x3 convolution into two stages: depth-wise convolution, which applies a single convolutional filter per input channel, followed by point-wise convolution that utilizes 1x1 convolutions for channel-wise combination. By segregating spatial and channel-wise information, MobileNet achieves a lightweight architecture without compromising accuracy, ideal for devices with limited resources.

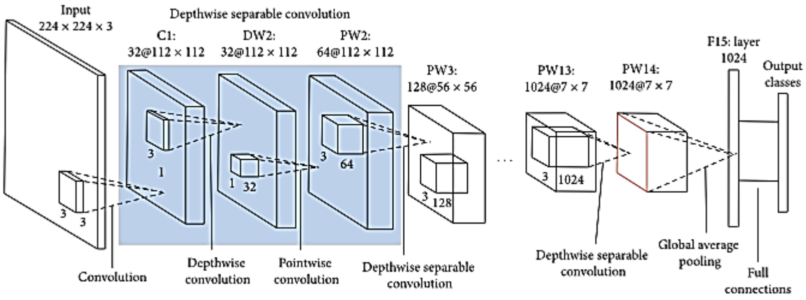


Figure 5: MobileNet Architecture (Andrew G. Howard et. al, 2017)

3.2.2 EfficientNet

EfficientNet, introduced by Mingxing Tan and Quoc V. Le in 2019, stands as a cutting-edge deep learning architecture characterized by its revolutionary compound scaling strategy. This approach systematically adjusts width, depth, and resolution to strike an optimal trade-off between model size and performance. The architecture is founded on repeating blocks of multiple convolutional layers, meticulously designed to capture hierarchical features. The innovative compound scaling empowers customization according to diverse resource constraints. EfficientNet's superiority over traditional models, with notably fewer resources, positions it as a highly efficient solution for tasks spanning image classification, object detection, and segmentation.

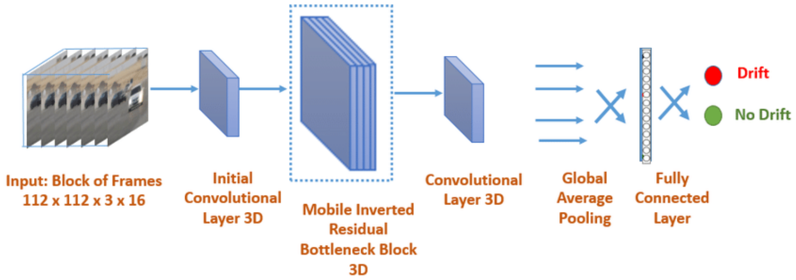


Figure 6: EfficientNet Architecture (Mingxing Tan and Quoc V. Le, 2019)

3.2.4 Resnet

ResNet50, devised by Kaiming He and colleagues in 2015, emerges as a transformative convolutional neural network architecture within the ResNet family. The architecture's pivotal innovation lies in its incorporation of residual blocks, enabling the learning of residual functions instead of direct mappings. Residual blocks encompass multiple convolutional layers accompanied by skip connections, which propagate input directly from one layer to a subsequent one. Featuring 50 layers, ResNet50 strategically employs varying convolutions, global average pooling, and fully connected layers for classification. The architecture's skip connections facilitate training of deeper networks without gradient vanishing, thereby elevating accuracy across diverse image recognition tasks.

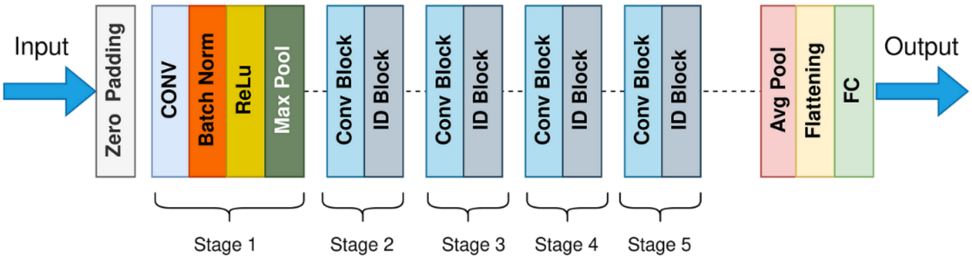


Figure 7: Resnet Architecture (Kaiming He et. al, 2015)

3.2.5 VGGNets

VGG19, an extension of VGG16, constitutes a convolutional neural network architecture renowned for its simplicity and consistency. The architecture encompasses 19 layers, including 16 convolutional and 3 fully connected layers. Characterized by 3x3 filters in convolutional layers and a consistent 224x224 input spatial resolution, VGG19 employs stacked convolutional and max-pooling layers, progressively diminishing spatial dimensions and augmenting network depth. The classification mechanism entails fully connected layers followed by SoftMax activation. VGG19's ease of implementation and impressive performance in image recognition, notably in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), solidified its prominence in the field.

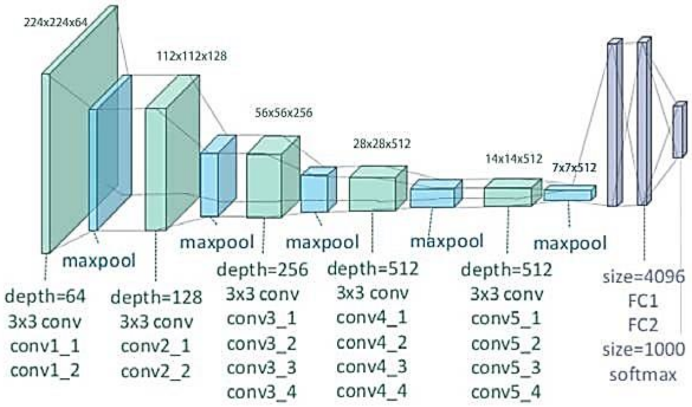


Figure 8: VGGNet Architecture

3.3 Research Method: Data Augmentation Techniques Used

3.3.1 Normalization

Data augmentation is a technique used in machine learning and deep learning, particularly in computer vision tasks. It involves creating new training data by applying various transformations to existing examples. These transformations include rotations, flips, translations, scaling, cropping, and changes in brightness, contrast, or color. Data augmentation aims to increase the diversity of the training dataset, which helps improve the model's ability to generalize and perform well on new, unseen data. By presenting the model with a wider range of variations, it learns to recognize features that are invariant to these changes, leading to enhanced robustness and reduced overfitting. Data augmentation is especially valuable when the available training data is limited, as it effectively increases the effective size of the dataset and can lead to more accurate and reliable machine learning models.

3.3.2 Random Eraser

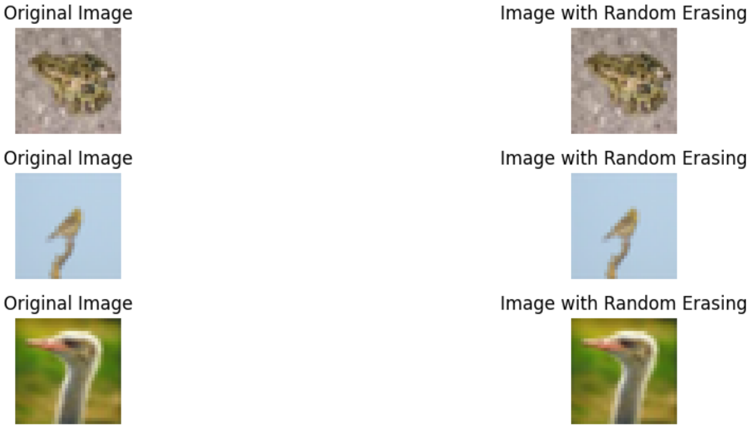

Original Image
Image with Random Erasing
Original Image
Image with Random Erasing
Original Image
Image with Random Erasing

Figure 9: Random Erasing example

The **`random_erasing`**(Kumar, T. 2022) function performs a data augmentation technique often used in computer vision tasks. It enhances neural network training by randomly erasing a rectangular region in an input image. The probability parameter controls the likelihood of applying the transformation. A target area and aspect ratio are randomly selected within specified ranges. The function then calculates the erasing rectangle's dimensions and position, ensuring they fit within the image. This region is replaced with random pixel values, simulating occlusion. This process encourages the network to learn more robust features and reduces overfitting. By introducing controlled variations, the function aims to improve the model's generalization performance during training, ultimately enhancing its ability to handle diverse real-world scenarios.

3.3.3 Saliency Gradient

Saliency functions enable the visualization of regions that significantly influence the output of a neural network, offering insights into how the model makes predictions. This type of data augmentation aids in understanding the model's decision-making process and identifying areas where its attention is focused.

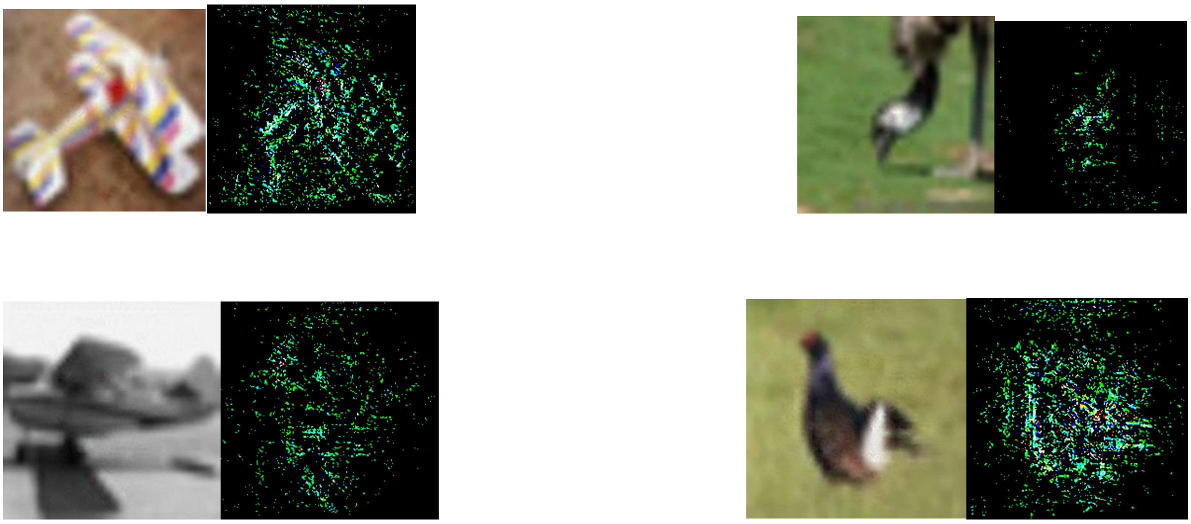
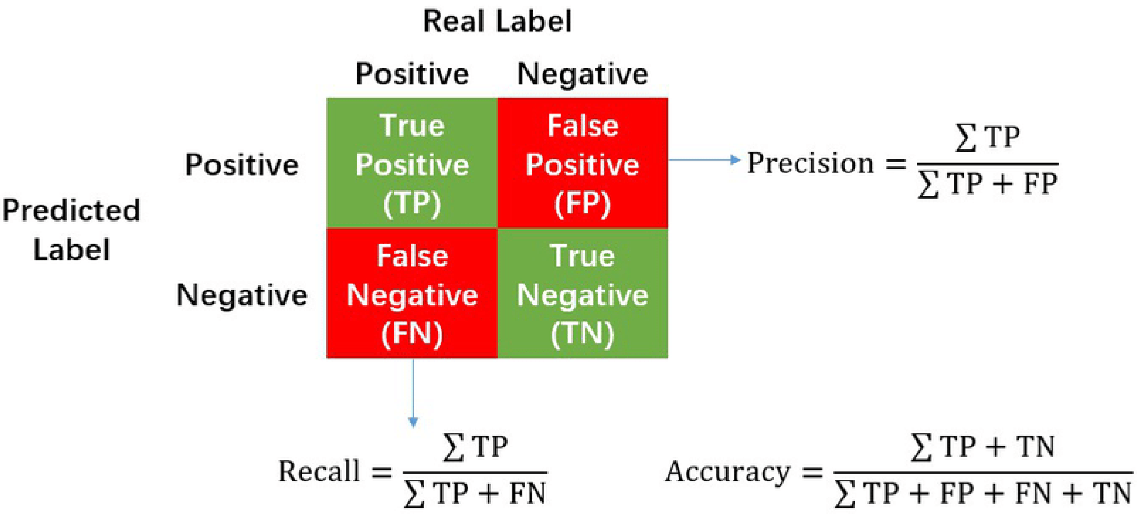


Figure 10: The left side are normal image while the right image is salient image.

3.4 Research Method: Evaluation



Real Label
Positive
Negative
Predicted Label
Positive
Negative
True Positive (TP)
False Positive (FP)
False Negative (FN)
True Negative (TN)
$\text{Precision} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FP}}$
$\text{Recall} = \frac{\sum \text{TP}}{\sum \text{TP} + \text{FN}}$
$\text{Accuracy} = \frac{\sum \text{TP} + \text{TN}}{\sum \text{TP} + \text{FP} + \text{FN} + \text{TN}}$

Figure 11: Confusion Matrix and different performance matrices formulas (Adhikari, N.D., 2020)

4 Design Specification

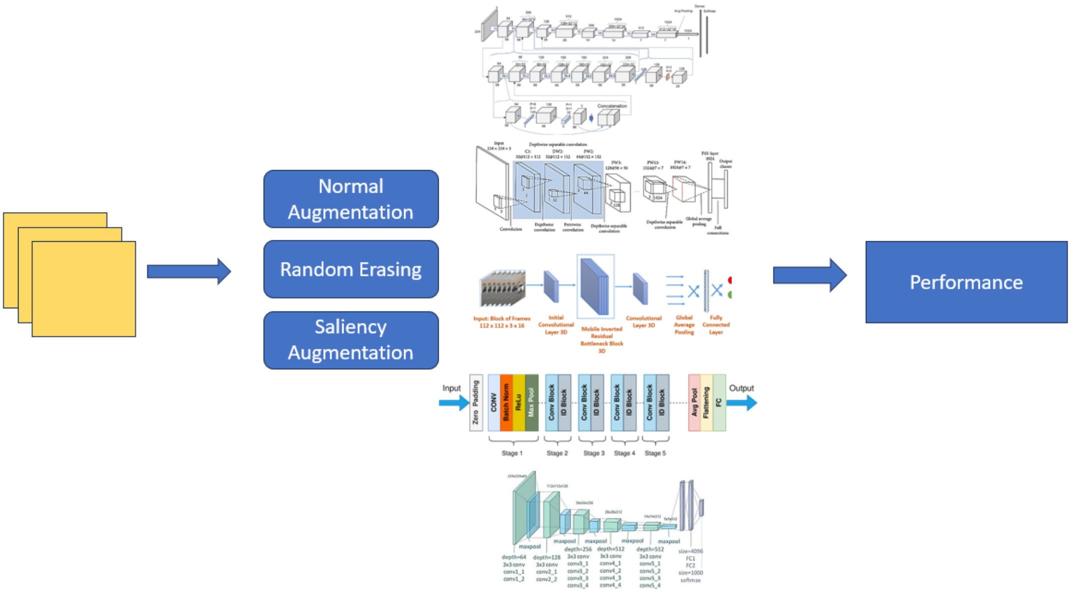


Figure 12: Network Architecture(Analytics Vidhya)

Algorithm Steps:

Step 1: Dataset Preparation: Obtain the CIFAR-10 dataset, which consists of 60,000 32x32 color images in 10 classes. Split the dataset into training and validation sets with appropriate proportions. We also use Fashion MNIST data set for validation.

Step 2: Augmentation Techniques: Implement three augmentation techniques: Normal Augmentation, Random Erasing, and Saliency Gradient Augmentation.

i. Normal Augmentation: Apply standard image transformations such as random flips, rotations, and brightness adjustments during training. Ensure that these augmentations maintain class labels.

ii. Random Erasing: Apply Random Erasing augmentation to randomly selected patches of the input images during training. Adjust parameters like erasing probability, size, and aspect ratio for optimal performance.

iii. Saliency Gradient Augmentation: Implement Saliency Gradient Augmentation by generating saliency maps for the training images. Combine the saliency maps with the original images to highlight important regions. Fine-tune parameters such as saliency threshold and blending factor.

Step 3: Model Training and Evaluation: Choose the VGG19, ResNet50, DenseNet121, MobileNetV2, and EfficientNetB0 architectures for training. For each model, establish a consistent training process:

i. Initialize the model's weights and compile it with a suitable optimizer and loss function.

ii. Train the model using the augmented training dataset.

iii. Monitor accuracy and loss on the validation set during training epochs.

Step 4: Performance Metrics: Track the following metrics across epochs:

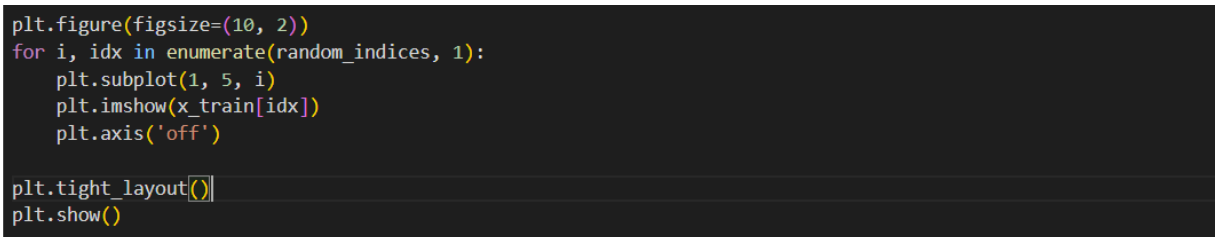
i. Accuracy: Measure the proportion of correctly classified validation images.
ii. Loss: Compute the average loss across validation set predictions.
iii. Confusion Matrix: After training, generate a confusion matrix for each model. Use the trained models to predict class labels for the validation dataset. Construct a confusion matrix that represents the true and predicted classes.

Step 4: Model Comparison and Analysis: Compare the accuracy and loss curves of each model for different augmentation techniques. Analyze the confusion matrices to identify any patterns of misclassifications or improved class separations.

Step 5: Conclusion and Insights: Draw conclusions about the effectiveness of Normal Augmentation, Random Erasing, and Saliency Gradient Augmentation on the performance of VGG19, ResNet50, DenseNet121, MobileNetV2, and EfficientNetB0 models on the CIFAR-10 dataset.

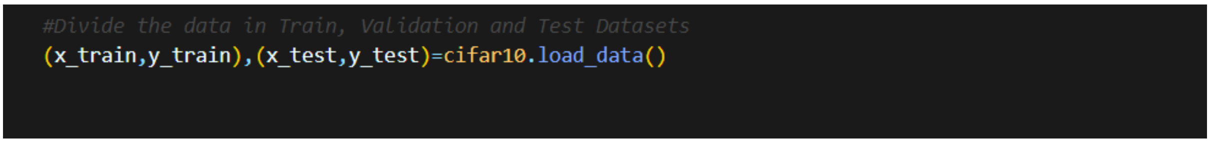
5 Implementation

5.1 Cifar10 dataset

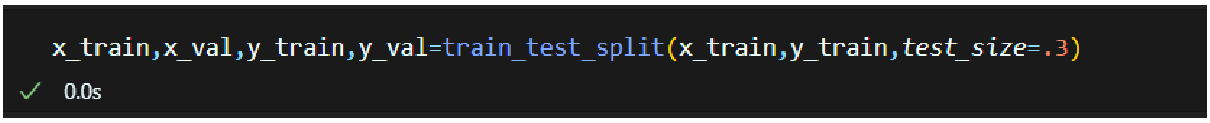


```
plt.figure(figsize=(10, 2))
for i, idx in enumerate(random_indices, 1):
    plt.subplot(1, 5, i)
    plt.imshow(x_train[idx])
    plt.axis('off')

plt.tight_layout()
plt.show()
```

5.2 Data Preprocessing



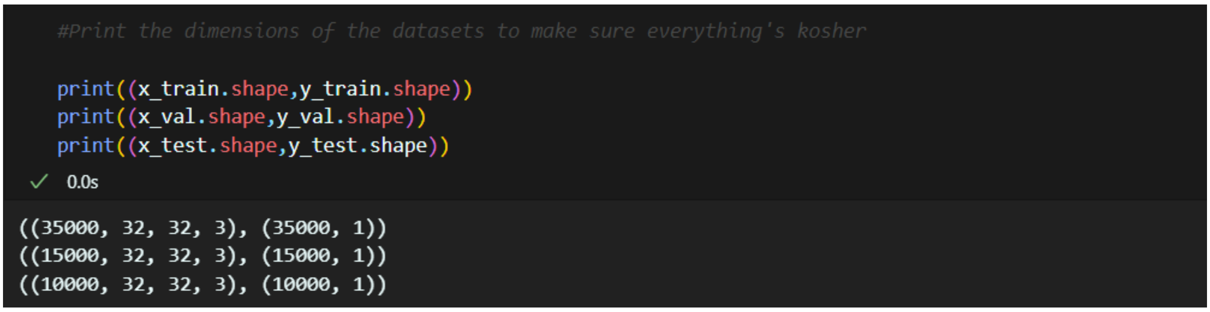
```
#Divide the data in Train, Validation and Test Datasets
(x_train,y_train),(x_test,y_test)=cifar10.load_data()
```



```
x_train,x_val,y_train,y_val=train_test_split(x_train,y_train,test_size=.3)
```

✓ 0.0s

Splitting training data for validation with 30% size to assess model training.

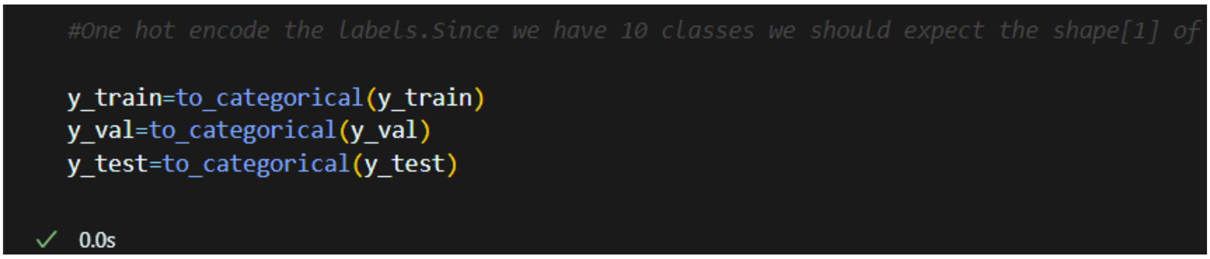


```
#Print the dimensions of the datasets to make sure everything's kosher

print((x_train.shape,y_train.shape))
print((x_val.shape,y_val.shape))
print((x_test.shape,y_test.shape))
```

✓ 0.0s

```
((35000, 32, 32, 3), (35000, 1))
((15000, 32, 32, 3), (15000, 1))
((10000, 32, 32, 3), (10000, 1))
```

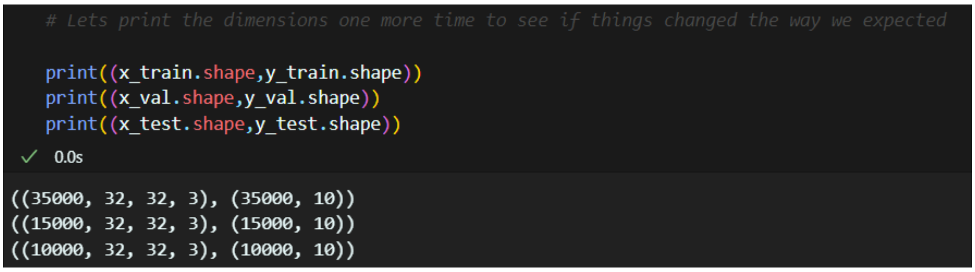
Code checks and displays data dimensions for training, validation, and testing sets.



```
#One hot encode the labels.Since we have 10 classes we should expect the shape[1] of

y_train=to_categorical(y_train)
y_val=to_categorical(y_val)
y_test=to_categorical(y_test)
```

✓ 0.0s

Converts categorical labels into one-hot encoded format, changing shape[1] from 1 to 10 for each y_train, y_val, and y_test dataset. This is used for multi-class classificatiossn.

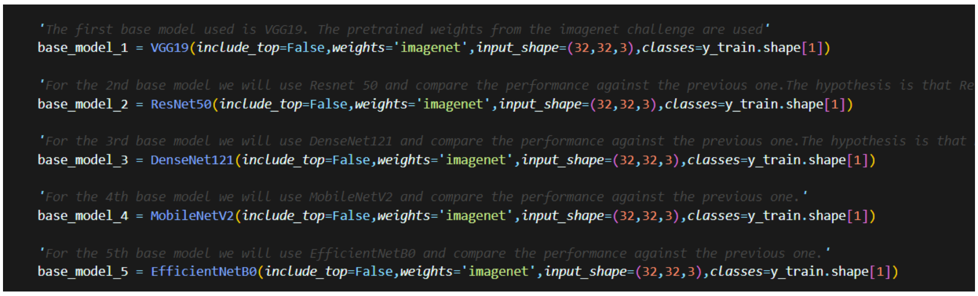


```
# Lets print the dimensions one more time to see if things changed the way we expected

print((x_train.shape,y_train.shape))
print((x_val.shape,y_val.shape))
print((x_test.shape,y_test.shape))
```

✓ 0.0s

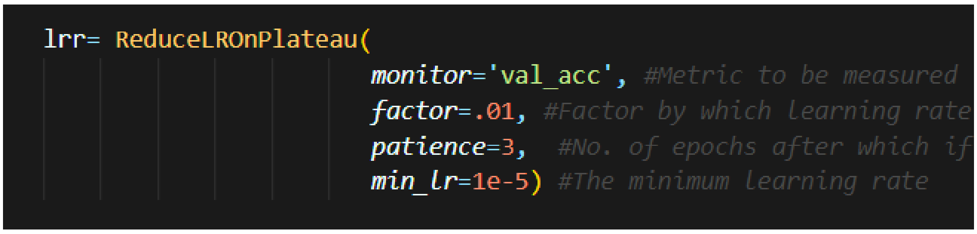
```
((35000, 32, 32, 3), (35000, 10))
((15000, 32, 32, 3), (15000, 10))
((10000, 32, 32, 3), (10000, 10))
```

5.3 Modelling Transfer Learning

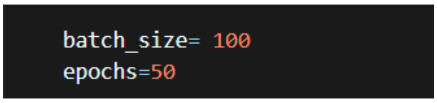


```
'The first base model used is VGG19. The pretrained weights from the imagenet challenge are used'
base_model_1 = VGG19(include_top=False,weights='imagenet',input_shape=(32,32,3),classes=y_train.shape[1])

'For the 2nd base model we will use Resnet 50 and compare the performance against the previous one.The hypothesis is that Re
base_model_2 = ResNet50(include_top=False,weights='imagenet',input_shape=(32,32,3),classes=y_train.shape[1])

'For the 3rd base model we will use DenseNet121 and compare the performance against the previous one.The hypothesis is that
base_model_3 = DenseNet121(include_top=False,weights='imagenet',input_shape=(32,32,3),classes=y_train.shape[1])

'For the 4th base model we will use MobileNetV2 and compare the performance against the previous one.'
base_model_4 = MobileNetV2(include_top=False,weights='imagenet',input_shape=(32,32,3),classes=y_train.shape[1])

'For the 5th base model we will use EfficientNetB0 and compare the performance against the previous one.'
base_model_5 = EfficientNetB0(include_top=False,weights='imagenet',input_shape=(32,32,3),classes=y_train.shape[1])
```

This code snippets defines all the model used in here.



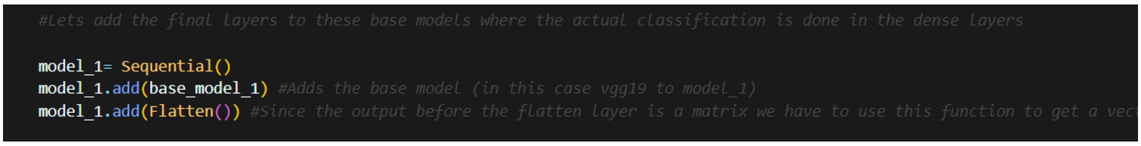
```
lrr= ReduceLROnPlateau(
                        monitor='val_acc', #Metric to be measured
                        factor=.01, #Factor by which learning rate
                        patience=3,  #No. of epochs after which if
                        min_lr=1e-5) #The minimum learning rate
```



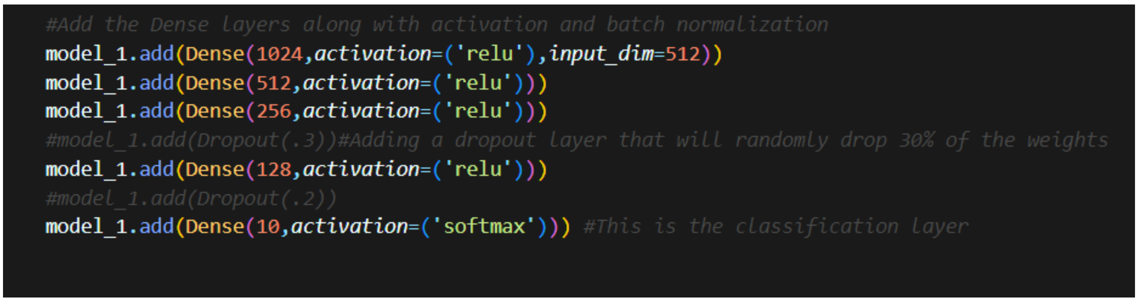
```
batch_size= 100
epochs=50
```

This code Snippet defines the **batch_size** and **epochs.** The code snippets below show the Layers used in the different base_models defined before.

5.3.1 VGG19

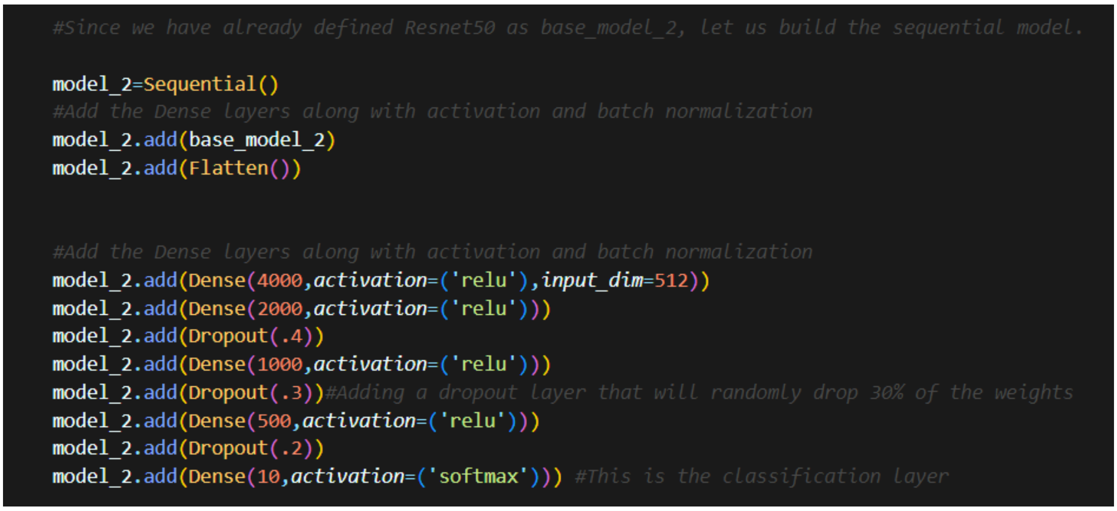


```
#Lets add the final layers to these base models where the actual classification is done in the dense layers

model_1= Sequential()
model_1.add(base_model_1) #Adds the base model (in this case vgg19 to model_1)
model_1.add(Flatten()) #Since the output before the flatten layer is a matrix we have to use this function to get a vec
```



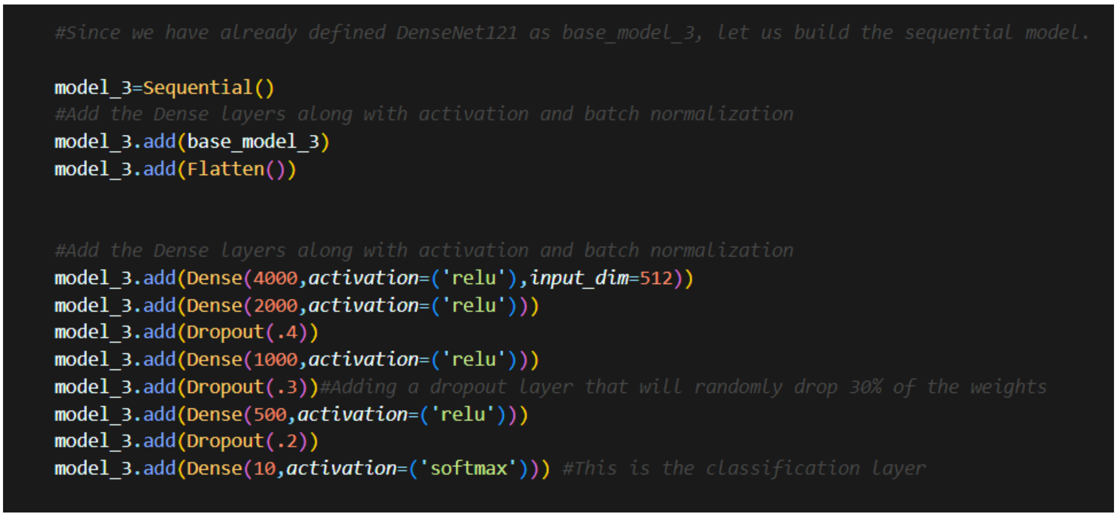
```
#Add the Dense layers along with activation and batch normalization
model_1.add(Dense(1024,activation=('relu'),input_dim=512))
model_1.add(Dense(512,activation=('relu')))
model_1.add(Dense(256,activation=('relu')))
#model_1.add(Dropout(.3))#Adding a dropout layer that will randomly drop 30% of the weights
model_1.add(Dense(128,activation=('relu')))
#model_1.add(Dropout(.2))
model_1.add(Dense(10,activation=('softmax'))) #This is the classification layer
```

5.3.2 ResNet50



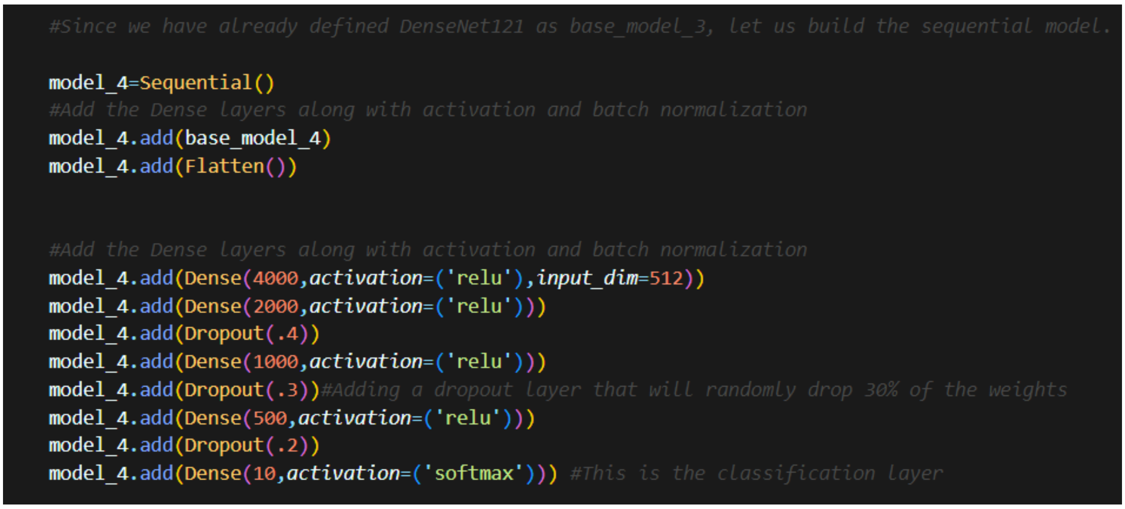
```
#Since we have already defined Resnet50 as base_model_2, let us build the sequential model.

model_2=Sequential()
#Add the Dense layers along with activation and batch normalization
model_2.add(base_model_2)
model_2.add(Flatten())


#Add the Dense layers along with activation and batch normalization
model_2.add(Dense(4000,activation=('relu'),input_dim=512))
model_2.add(Dense(2000,activation=('relu')))
model_2.add(Dropout(.4))
model_2.add(Dense(1000,activation=('relu')))
model_2.add(Dropout(.3))#Adding a dropout layer that will randomly drop 30% of the weights
model_2.add(Dense(500,activation=('relu')))
model_2.add(Dropout(.2))
model_2.add(Dense(10,activation=('softmax'))) #This is the classification layer
```

5.3.3 DenseNet121



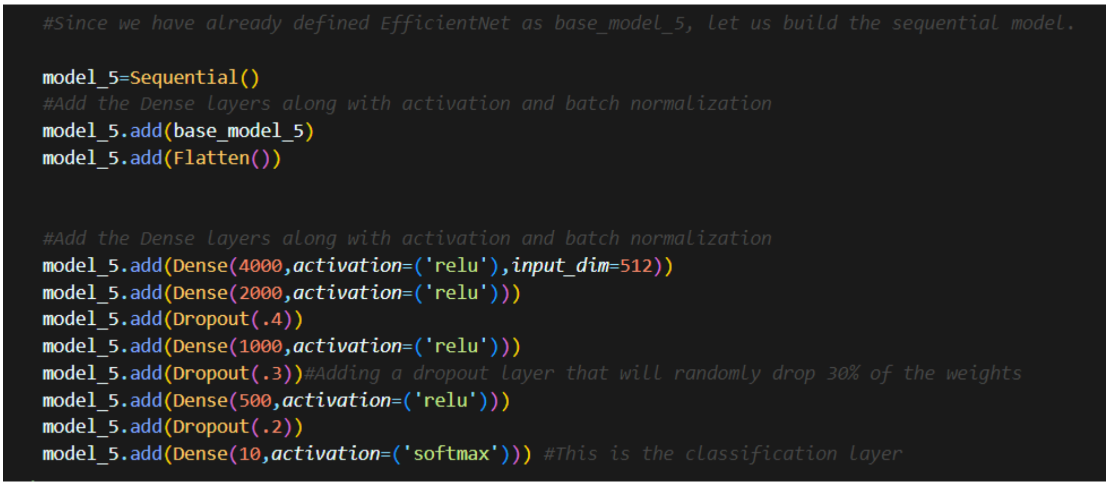
```
#Since we have already defined DenseNet121 as base_model_3, let us build the sequential model.

model_3=Sequential()
#Add the Dense layers along with activation and batch normalization
model_3.add(base_model_3)
model_3.add(Flatten())


#Add the Dense layers along with activation and batch normalization
model_3.add(Dense(4000,activation=('relu'),input_dim=512))
model_3.add(Dense(2000,activation=('relu')))
model_3.add(Dropout(.4))
model_3.add(Dense(1000,activation=('relu')))
model_3.add(Dropout(.3))#Adding a dropout layer that will randomly drop 30% of the weights
model_3.add(Dense(500,activation=('relu')))
model_3.add(Dropout(.2))
model_3.add(Dense(10,activation=('softmax'))) #This is the classification layer
```

5.3.4 MobileNetV2



```
#Since we have already defined DenseNet121 as base_model_3, let us build the sequential model.

model_4=Sequential()
#Add the Dense layers along with activation and batch normalization
model_4.add(base_model_4)
model_4.add(Flatten())


#Add the Dense layers along with activation and batch normalization
model_4.add(Dense(4000,activation=('relu'),input_dim=512))
model_4.add(Dense(2000,activation=('relu')))
model_4.add(Dropout(.4))
model_4.add(Dense(1000,activation=('relu')))
model_4.add(Dropout(.3))#Adding a dropout layer that will randomly drop 30% of the weights
model_4.add(Dense(500,activation=('relu')))
model_4.add(Dropout(.2))
model_4.add(Dense(10,activation=('softmax'))) #This is the classification layer
```

5.3.5 EfficientNet



```
#Since we have already defined EfficientNet as base_model_5, let us build the sequential model.

model_5=Sequential()
#Add the Dense layers along with activation and batch normalization
model_5.add(base_model_5)
model_5.add(Flatten())


#Add the Dense layers along with activation and batch normalization
model_5.add(Dense(4000,activation=('relu'),input_dim=512))
model_5.add(Dense(2000,activation=('relu')))
model_5.add(Dropout(.4))
model_5.add(Dense(1000,activation=('relu')))
model_5.add(Dropout(.3))#Adding a dropout layer that will randomly drop 30% of the weights
model_5.add(Dense(500,activation=('relu')))
model_5.add(Dropout(.2))
model_5.add(Dense(10,activation=('softmax'))) #This is the classification layer
```

6 Results

Case1: Normal Data Augmentation

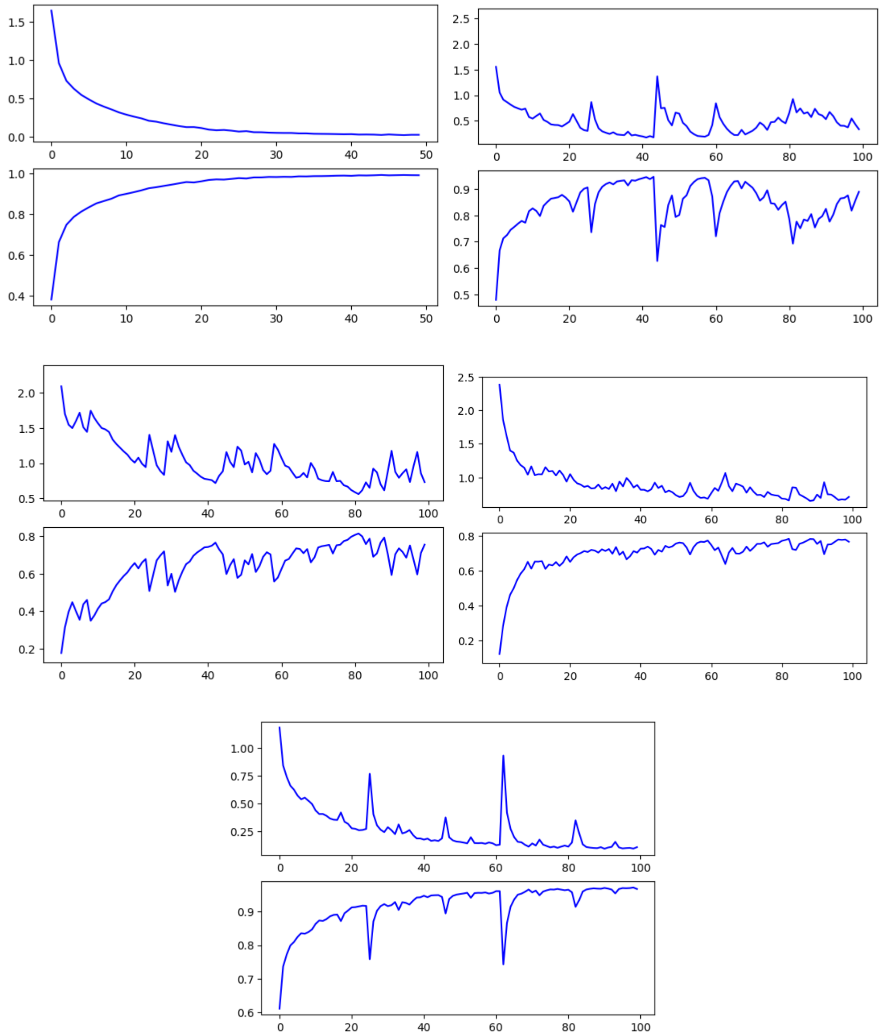
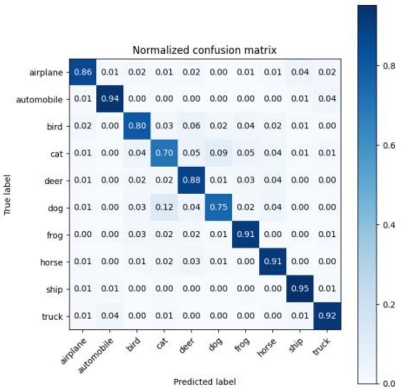
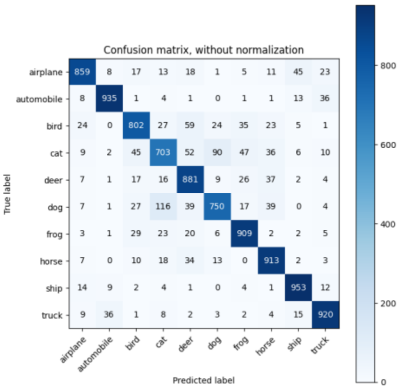
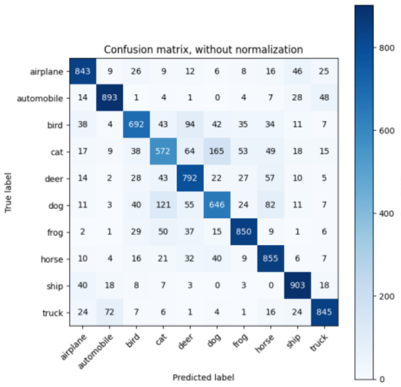


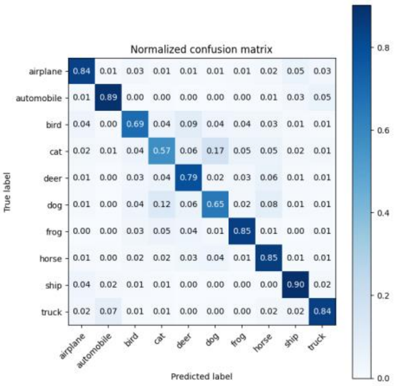
Figure 13: Performance Curve of VGGNet19, ResNet50, DenseNet121, MobileNetV2 and EfficientNetB0 using Normal Augmentation

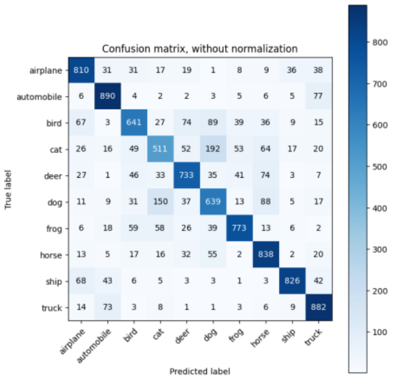
Indications of successful training of VGG19 include a decreasing loss and an increasing accuracy per epoch. ResNet50's results show that accuracy increases with training, but a rising loss rate suggests overfitting. Overfitting but also improved generalisation may be behind DenseNet121's increasing accuracy and decreasing loss up to a plateau at around 60 epochs. MobileNet suffers from underfitting, which leads to less accurate predictions and more loss. This indicates that the model is oversimplified. Declining accuracy and rising loss are symptoms of overfitting in EfficientNet; stopping early or using regularisation might assist reduce unnecessary complexity. Models' inconsistent behaviour show the need for vigilant monitoring and fine-tuning of training techniques for best results and generalisation.

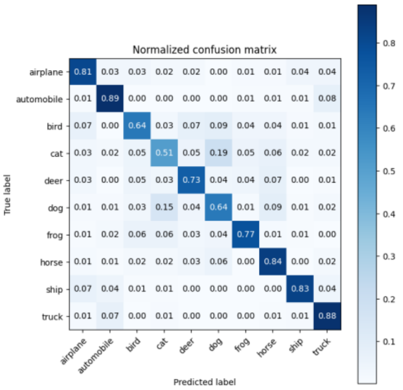

Normalized confusion matrix
airplane 0.86 0.01 0.02 0.01 0.02 0.00 0.01 0.01 0.04 0.02
automobile 0.01 0.94 0.00 0.00 0.00 0.00 0.00 0.00 0.01 0.04
bird 0.02 0.00 0.80 0.03 0.06 0.02 0.04 0.02 0.01 0.00
cat 0.01 0.00 0.04 0.70 0.05 0.09 0.05 0.04 0.01 0.01
deer 0.01 0.00 0.02 0.02 0.88 0.01 0.03 0.04 0.00 0.00
dog 0.01 0.00 0.03 0.12 0.04 0.75 0.02 0.04 0.00 0.00
frog 0.00 0.00 0.03 0.02 0.02 0.01 0.91 0.00 0.00 0.01
horse 0.01 0.00 0.01 0.02 0.03 0.01 0.00 0.91 0.00 0.00
ship 0.01 0.01 0.00 0.00 0.00 0.00 0.00 0.00 0.95 0.01
truck 0.01 0.04 0.00 0.01 0.00 0.00 0.00 0.00 0.01 0.92
True label
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Predicted label
0.8
0.6
0.4
0.2
0.0

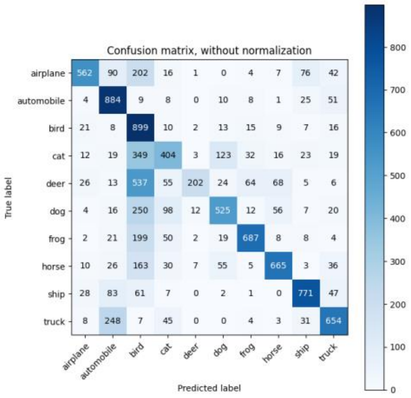

Confusion matrix, without normalization
airplane 859 8 17 13 18 1 5 11 45 23
automobile 8 935 1 4 1 0 1 1 13 36
bird 24 0 802 27 59 24 35 23 5 1
cat 9 2 45 703 52 90 47 36 6 10
deer 7 1 17 16 881 9 26 37 2 4
dog 7 1 27 116 39 750 17 39 0 4
frog 3 1 29 23 20 6 909 2 2 5
horse 7 0 10 18 34 13 0 913 2 3
ship 14 9 2 4 1 0 4 1 953 12
truck 9 36 1 8 2 3 2 4 15 920
True label
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Predicted label
800
600
400
200
0

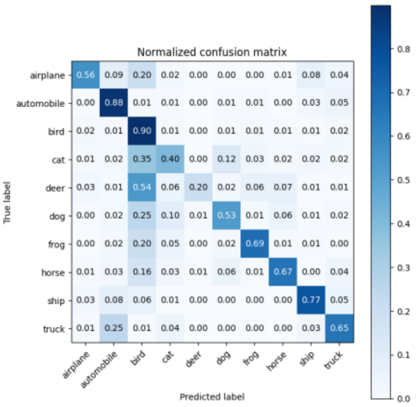

Confusion matrix, without normalization
airplane 843 9 26 9 12 6 8 16 46 25
automobile 14 893 1 4 1 0 4 7 28 48
bird 38 4 692 43 94 42 35 34 11 7
cat 17 9 38 572 64 165 53 49 18 15
deer 14 2 28 43 792 22 27 57 10 5
dog 11 3 40 121 55 646 24 82 11 7
frog 2 1 29 50 37 15 850 9 1 6
horse 10 4 16 21 32 40 9 855 6 7
ship 40 18 8 7 3 0 3 0 903 18
truck 24 72 7 6 1 4 1 16 24 845
True label
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Predicted label
800
600
400
200
0


Normalized confusion matrix
airplane 0.84 0.01 0.03 0.01 0.01 0.01 0.01 0.02 0.05 0.03
automobile 0.01 0.89 0.00 0.00 0.00 0.00 0.00 0.01 0.03 0.05
bird 0.04 0.00 0.69 0.04 0.09 0.04 0.04 0.03 0.01 0.01
cat 0.02 0.01 0.04 0.57 0.06 0.17 0.05 0.05 0.02 0.01
deer 0.01 0.00 0.03 0.04 0.79 0.02 0.03 0.06 0.01 0.01
dog 0.01 0.00 0.04 0.12 0.06 0.65 0.02 0.08 0.01 0.01
frog 0.00 0.00 0.03 0.05 0.04 0.01 0.85 0.01 0.00 0.01
horse 0.01 0.00 0.02 0.02 0.03 0.04 0.01 0.85 0.01 0.01
ship 0.04 0.02 0.01 0.01 0.00 0.00 0.00 0.00 0.90 0.02
truck 0.02 0.07 0.01 0.01 0.00 0.00 0.00 0.02 0.02 0.84
True label
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Predicted label
0.8
0.6
0.4
0.2
0.0


Confusion matrix, without normalization
airplane 810 31 31 17 19 1 8 9 36 38
automobile 6 890 4 2 2 3 5 6 5 77
bird 67 3 641 27 74 89 39 36 9 15
cat 26 16 49 511 52 192 53 64 17 20
deer 27 1 46 33 733 35 41 74 3 7
dog 11 9 31 150 37 639 13 88 5 17
frog 6 18 59 58 26 39 773 13 6 2
horse 13 5 17 16 32 55 2 838 2 20
ship 68 43 6 5 3 3 1 3 826 42
truck 14 73 3 8 1 1 3 6 9 882
True label
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Predicted label
800
700
600
500
400
300
200
100


Normalized confusion matrix
airplane 0.81 0.03 0.03 0.02 0.02 0.00 0.01 0.01 0.04 0.04
automobile 0.01 0.89 0.00 0.00 0.00 0.00 0.01 0.01 0.01 0.08
bird 0.07 0.00 0.64 0.03 0.07 0.09 0.04 0.04 0.01 0.01
cat 0.03 0.02 0.05 0.51 0.05 0.19 0.05 0.06 0.02 0.02
deer 0.03 0.00 0.05 0.03 0.73 0.04 0.04 0.07 0.00 0.01
dog 0.01 0.01 0.03 0.15 0.04 0.64 0.01 0.09 0.01 0.02
frog 0.01 0.02 0.06 0.06 0.03 0.04 0.77 0.01 0.01 0.00
horse 0.01 0.01 0.02 0.02 0.03 0.06 0.00 0.84 0.00 0.02
ship 0.07 0.04 0.01 0.01 0.00 0.00 0.00 0.00 0.83 0.04
truck 0.01 0.07 0.00 0.01 0.00 0.00 0.00 0.01 0.01 0.88
True label
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Predicted label
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1


Confusion matrix, without normalization
airplane 562 90 202 16 1 0 4 7 76 42
automobile 4 884 9 8 0 10 8 1 25 51
bird 21 8 899 10 2 13 15 9 7 16
cat 12 19 349 404 3 123 32 16 23 19
deer 26 13 537 55 202 24 64 68 5 6
dog 4 16 250 98 12 525 12 56 7 20
frog 2 21 199 50 2 19 687 8 8 4
horse 10 26 163 30 7 55 5 665 3 36
ship 28 83 61 7 0 2 1 0 771 47
truck 8 248 7 45 0 0 4 3 31 654
True label
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Predicted label
800
700
600
500
400
300
200
100
0


Normalized confusion matrix
airplane 0.56 0.09 0.20 0.02 0.00 0.00 0.00 0.01 0.08 0.04
automobile 0.00 0.88 0.01 0.01 0.00 0.01 0.01 0.00 0.03 0.05
bird 0.02 0.01 0.90 0.01 0.00 0.01 0.01 0.01 0.01 0.02
cat 0.01 0.02 0.35 0.40 0.00 0.12 0.03 0.02 0.02 0.02
deer 0.03 0.01 0.54 0.06 0.20 0.02 0.06 0.07 0.01 0.01
dog 0.00 0.02 0.25 0.10 0.01 0.53 0.01 0.06 0.01 0.02
frog 0.00 0.02 0.20 0.05 0.00 0.02 0.69 0.01 0.01 0.00
horse 0.01 0.03 0.16 0.03 0.01 0.06 0.01 0.67 0.00 0.04
ship 0.03 0.08 0.06 0.01 0.00 0.00 0.00 0.00 0.77 0.05
truck 0.01 0.25 0.01 0.04 0.00 0.00 0.00 0.00 0.03 0.65
True label
airplane
automobile
bird
cat
deer
dog
frog
horse
ship
truck
Predicted label
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0

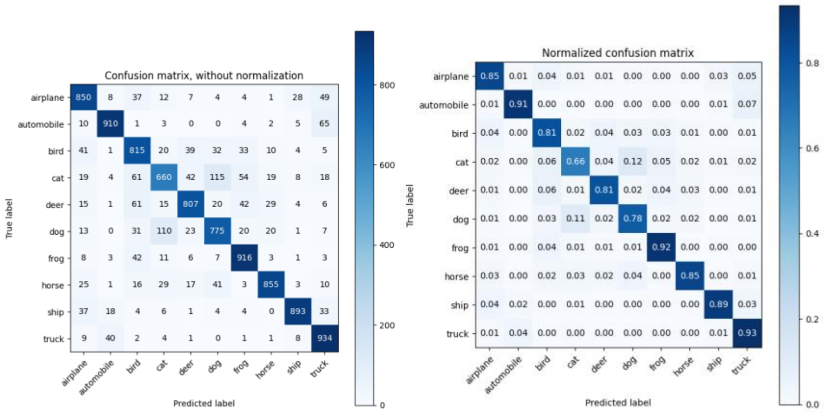
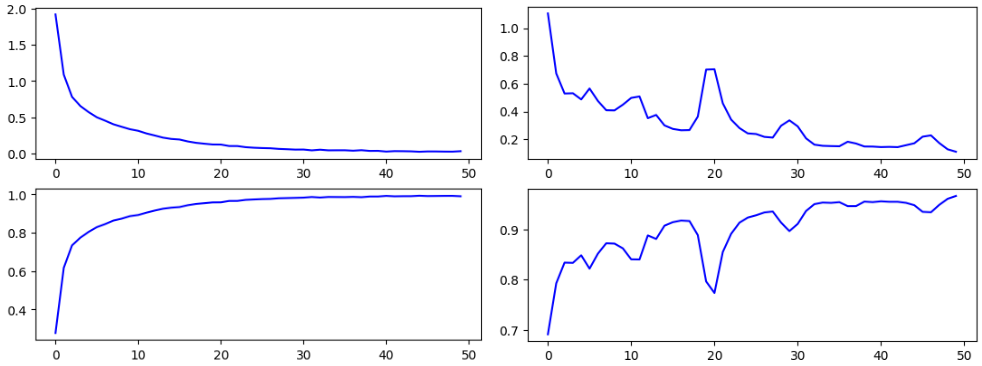
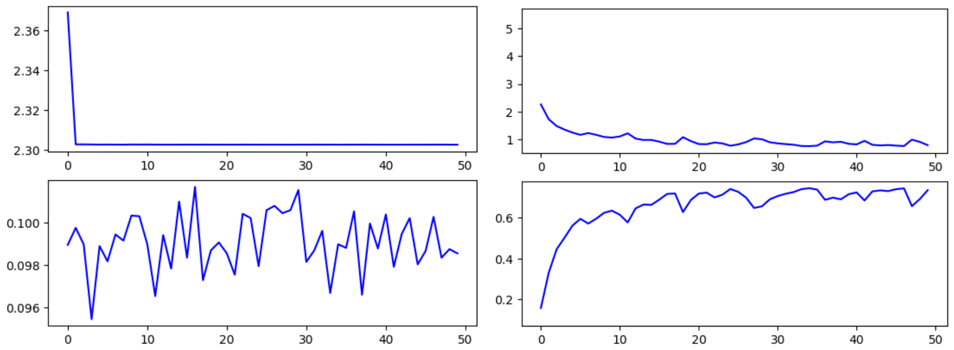


Figure 14: Confusion Matrix of VGGNet19, ResNet50, DenseNet121, MobileNetV2 and EfficientNetB0 using Normal Augmentation

VGG19 was chosen as the best model because to its superior classification performance, as seen by the diagonal dark gradient in the confusion matrix. Mistakes in DenseNet121 are grouped in the same area of the matrix as they are in ResNet50. Overall, ResNet50 performs well, with a few exceptions worth pointing out. MobileNetV2 has an unfavourable categorization, especially in its propensity to predict the "Bird" class. EfficientNetB0 shares several similarities with other matrices, such MobileNet's. These results draw attention to the superior classification offered by VGG19, the robust performance offered by ResNet50, the unique challenges posed by DenseNet121 and MobileNetV2, and the comparative alignment offered by EfficientNetB0.

Case2: Random Erasing Image





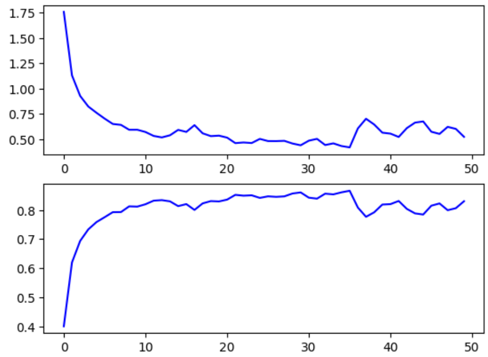
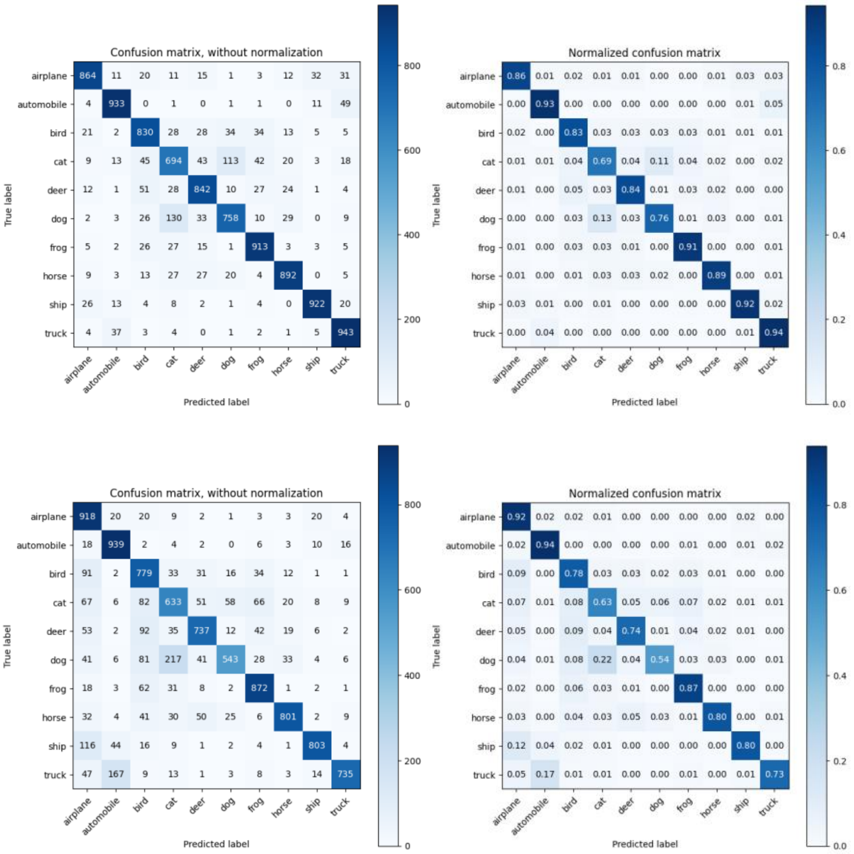


Figure 15: Performance Curve of VGGNet19, ResNet50, DenseNet121, MobileNetV2 and EfficientNetB0 for Random Erasing

The training progress shown by VGG19 is encouraging, as evidenced by a declining loss and a rising accuracy per epoch. A convergence in accuracy and loss at 40 epochs is seen by ResNet50, which is suggestive of learning saturation. The findings that DenseNet121 produces are less than ideal, with a maximum accuracy of 0.1271. MobileNetV2 exhibits balanced learning since its accuracy and loss levels reach a plateau after around 40 epochs, which is indicative of regulated generalisation. EfficientNetB0 is a model that demonstrates progressive learning, which involves improving predictions with increasing epochs while avoiding overfitting. This confirms that the model is able to effectively adapt to training data.



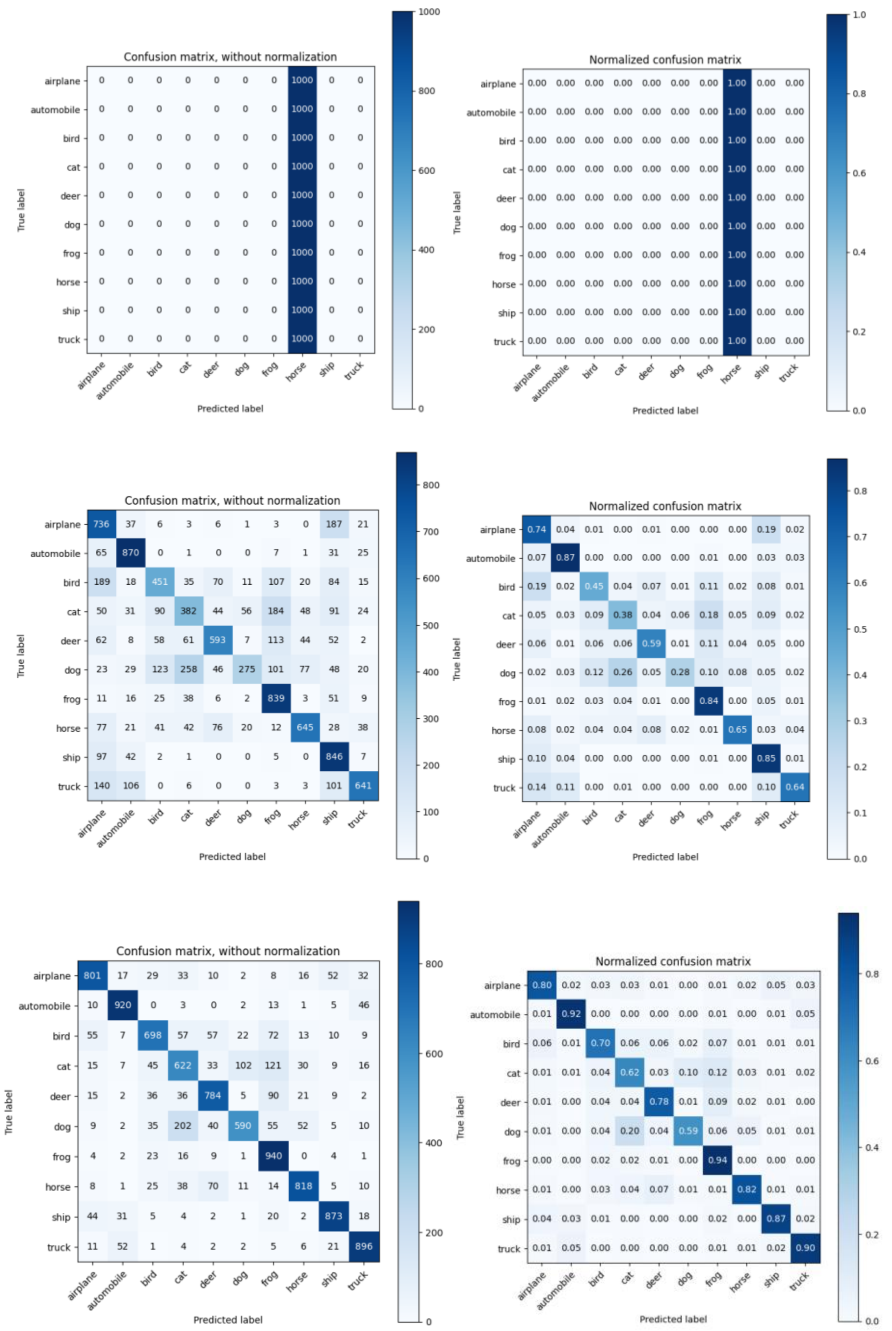


Figure 16: Confusion Matrix of VGGNet19, ResNet50, DenseNet121, MobileNetV2 and EfficientNetB0 for Random Erasing

The result this offers is second to vgg19 one.

Case3: Salience Gradient Based Augmentation

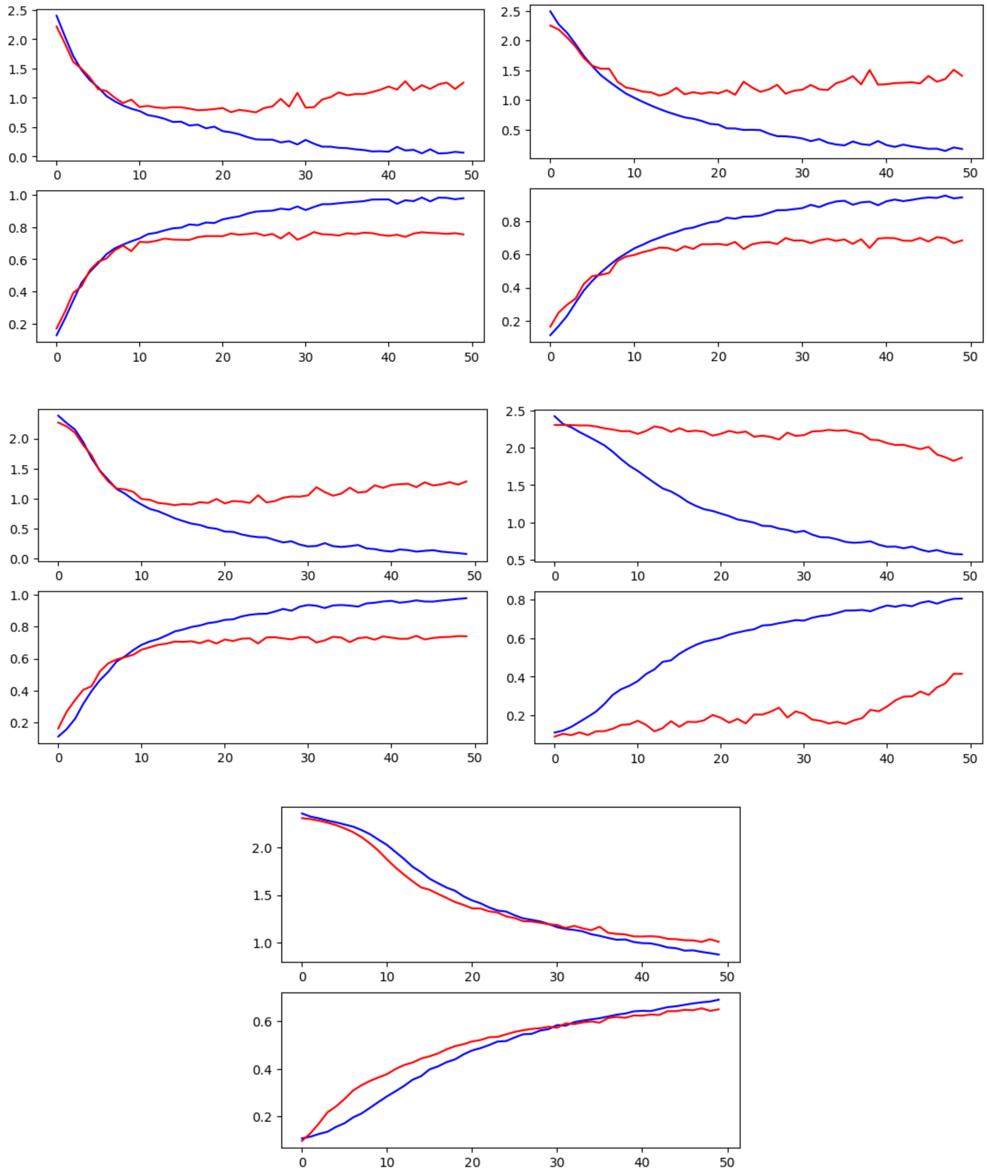
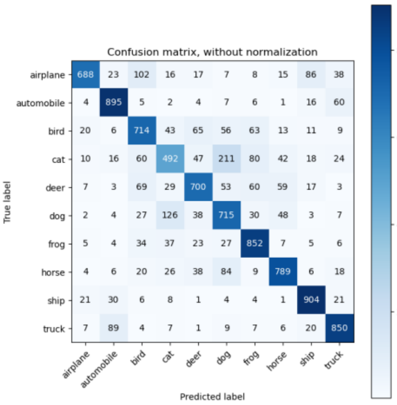
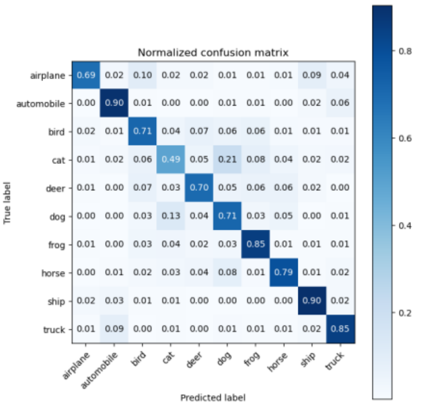
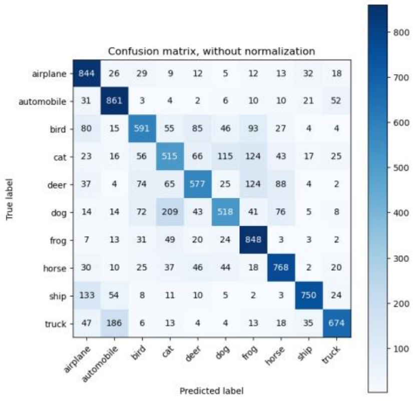


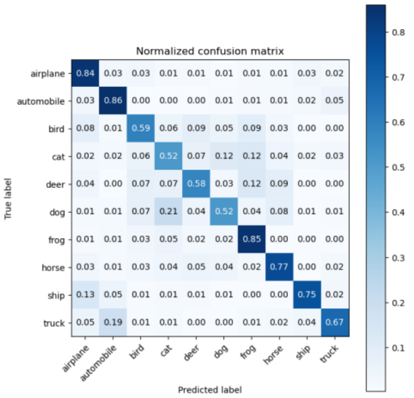
Figure 17: Performance Curve of VGGNet19, ResNet50, DenseNet121, MobileNetV2 and EfficientNetB0 for Saliency Gradient

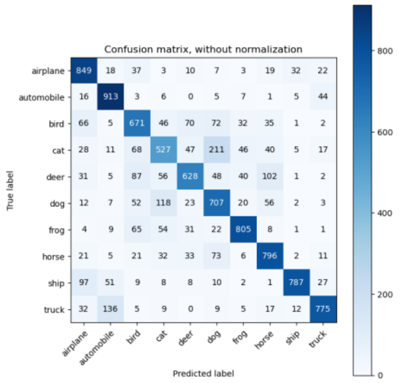
The results of both the training and validation phases show that VGGNet has excellent performance. ResNet has a considerable decrease in loss, which suggests improved performance, while still preserving a constant learning rate. Because the training and validation accuracies are different, there is a risk of a very minor overfitting occurring. The results of DenseNet121's validation show that it is getting more accurate while simultaneously reducing its loss, which might point to overfitting problems. Around 30 epochs is when MobileNetV2 reaches its maximum performance, which is characterised by increasing accuracy and decreasing loss; further training increases the likelihood of overfitting. The EfficientNetB0 training loss reduction is in line with the steady accuracy gain, but the restricted validation accuracy requires cautious modifications to the learning rate due to the absence of measurements.

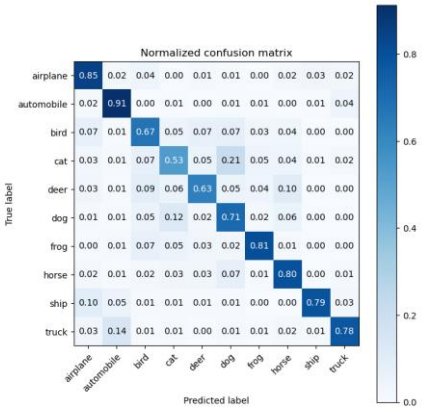

Confusion matrix, without normalization
airplane 688 23 102 16 17 7 8 15 86 38
automobile 4 895 5 2 4 7 6 1 16 60
bird 20 6 714 43 65 56 63 13 11 9
cat 10 16 60 492 47 211 80 42 18 24
deer 7 3 69 29 700 53 60 59 17 3
dog 2 4 27 126 38 715 30 48 3 7
frog 5 4 34 37 23 27 852 7 5 6
horse 4 6 20 26 38 84 9 789 6 18
ship 21 30 6 8 1 4 4 1 904 21
truck 7 89 4 7 1 9 7 6 20 850
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
800 600 400 200

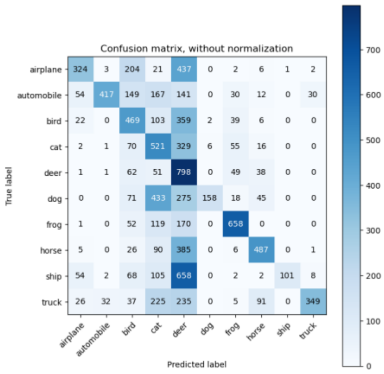

Normalized confusion matrix
airplane 0.69 0.02 0.10 0.02 0.02 0.01 0.01 0.01 0.09 0.04
automobile 0.00 0.90 0.01 0.00 0.00 0.01 0.01 0.00 0.02 0.06
bird 0.02 0.01 0.71 0.04 0.07 0.06 0.06 0.01 0.01 0.01
cat 0.01 0.02 0.06 0.49 0.05 0.21 0.08 0.04 0.02 0.02
deer 0.01 0.00 0.07 0.03 0.70 0.05 0.06 0.06 0.02 0.00
dog 0.00 0.00 0.03 0.13 0.04 0.71 0.03 0.05 0.00 0.01
frog 0.01 0.00 0.03 0.04 0.02 0.03 0.85 0.01 0.01 0.01
horse 0.00 0.01 0.02 0.03 0.04 0.08 0.01 0.79 0.01 0.02
ship 0.02 0.03 0.01 0.01 0.00 0.00 0.00 0.00 0.90 0.02
truck 0.01 0.09 0.00 0.01 0.00 0.01 0.01 0.01 0.02 0.85
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
0.8 0.6 0.4 0.2

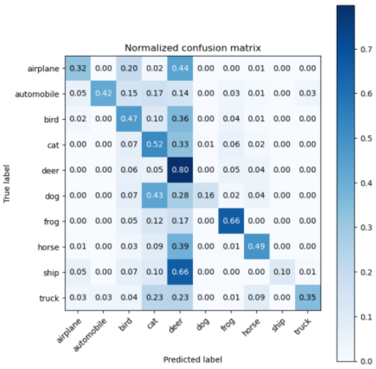

Confusion matrix, without normalization
airplane 844 26 29 9 12 5 12 13 32 18
automobile 31 861 3 4 2 6 10 10 21 52
bird 80 15 591 55 85 46 93 27 4 4
cat 23 16 56 515 66 115 124 43 17 25
deer 37 4 74 65 577 25 124 88 4 2
dog 14 14 72 209 43 518 41 76 5 8
frog 7 13 31 49 20 24 848 3 3 2
horse 30 10 25 37 46 44 18 768 2 20
ship 133 54 8 11 10 5 2 3 750 24
truck 47 186 6 13 4 4 13 18 35 674
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
800 700 600 500 400 300 200 100


Normalized confusion matrix
airplane 0.84 0.03 0.03 0.01 0.01 0.01 0.01 0.01 0.03 0.02
automobile 0.03 0.86 0.00 0.00 0.00 0.01 0.01 0.01 0.02 0.05
bird 0.08 0.01 0.59 0.06 0.09 0.05 0.09 0.03 0.00 0.00
cat 0.02 0.02 0.06 0.52 0.07 0.12 0.12 0.04 0.02 0.03
deer 0.04 0.00 0.07 0.07 0.58 0.03 0.12 0.09 0.00 0.00
dog 0.01 0.01 0.07 0.21 0.04 0.52 0.04 0.08 0.01 0.01
frog 0.01 0.01 0.03 0.05 0.02 0.02 0.85 0.00 0.00 0.00
horse 0.03 0.01 0.03 0.04 0.05 0.04 0.02 0.77 0.00 0.02
ship 0.13 0.05 0.01 0.01 0.01 0.01 0.00 0.00 0.75 0.02
truck 0.05 0.19 0.01 0.01 0.00 0.00 0.01 0.02 0.04 0.67
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
0.8 0.7 0.6 0.5 0.4 0.3 0.2 0.1


Confusion matrix, without normalization
airplane 849 18 37 3 10 7 3 19 32 22
automobile 16 913 3 6 0 5 7 1 5 44
bird 66 5 671 46 70 72 32 35 1 2
cat 28 11 68 527 47 211 46 40 5 17
deer 31 5 87 56 628 48 40 102 1 2
dog 12 7 52 118 23 707 20 56 2 3
frog 4 9 65 54 31 22 805 8 1 1
horse 21 5 21 32 33 73 6 796 2 11
ship 97 51 9 8 8 10 2 1 787 27
truck 32 136 5 9 0 9 5 17 12 775
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
800 600 400 200 0


Normalized confusion matrix
airplane 0.85 0.02 0.04 0.00 0.01 0.01 0.00 0.02 0.03 0.02
automobile 0.02 0.91 0.00 0.01 0.00 0.01 0.01 0.00 0.01 0.04
bird 0.07 0.01 0.67 0.05 0.07 0.07 0.03 0.04 0.00 0.00
cat 0.03 0.01 0.07 0.53 0.05 0.21 0.05 0.04 0.01 0.02
deer 0.03 0.01 0.09 0.06 0.63 0.05 0.04 0.10 0.00 0.00
dog 0.01 0.01 0.05 0.12 0.02 0.71 0.02 0.06 0.00 0.00
frog 0.00 0.01 0.07 0.05 0.03 0.02 0.81 0.01 0.00 0.00
horse 0.02 0.01 0.02 0.03 0.03 0.07 0.01 0.80 0.00 0.01
ship 0.10 0.05 0.01 0.01 0.01 0.01 0.00 0.00 0.79 0.03
truck 0.03 0.14 0.01 0.01 0.00 0.01 0.01 0.02 0.01 0.78
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
0.8 0.6 0.4 0.2 0.0


Confusion matrix, without normalization
airplane 324 3 204 21 437 0 2 6 1 2
automobile 54 417 149 167 141 0 30 12 0 30
bird 22 0 469 103 359 2 39 6 0 0
cat 2 1 70 521 329 6 55 16 0 0
deer 1 1 62 51 798 0 49 38 0 0
dog 0 0 71 433 275 158 18 45 0 0
frog 1 0 52 119 170 0 658 0 0 0
horse 5 0 26 90 385 0 6 487 0 1
ship 54 2 68 105 658 0 2 2 101 8
truck 26 32 37 225 235 0 5 91 0 349
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
700 600 500 400 300 200 100 0


Normalized confusion matrix
airplane 0.32 0.00 0.20 0.02 0.44 0.00 0.00 0.01 0.00 0.00
automobile 0.05 0.42 0.15 0.17 0.14 0.00 0.03 0.01 0.00 0.03
bird 0.02 0.00 0.47 0.10 0.36 0.00 0.04 0.01 0.00 0.00
cat 0.00 0.00 0.07 0.52 0.33 0.01 0.06 0.02 0.00 0.00
deer 0.00 0.00 0.06 0.05 0.80 0.00 0.05 0.04 0.00 0.00
dog 0.00 0.00 0.07 0.43 0.28 0.16 0.02 0.04 0.00 0.00
frog 0.00 0.00 0.05 0.12 0.17 0.00 0.66 0.00 0.00 0.00
horse 0.01 0.00 0.03 0.09 0.39 0.00 0.01 0.49 0.00 0.00
ship 0.05 0.00 0.07 0.10 0.66 0.00 0.00 0.00 0.10 0.01
truck 0.03 0.03 0.04 0.23 0.23 0.00 0.01 0.09 0.00 0.35
True label
airplane automobile bird cat deer dog frog horse ship truck
Predicted label
0.7 0.6 0.5 0.4 0.3 0.2 0.1 0.0

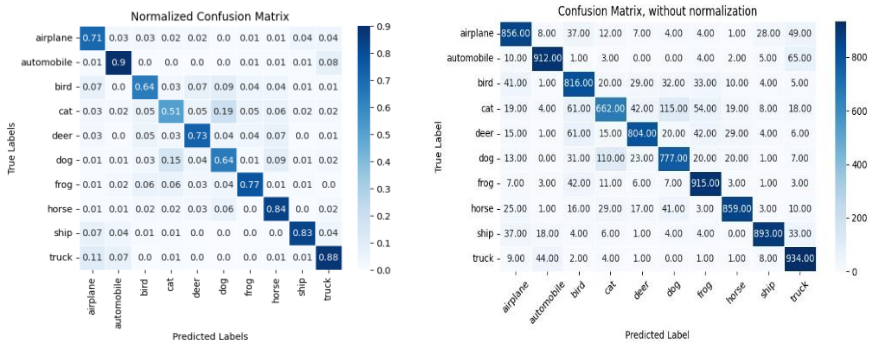
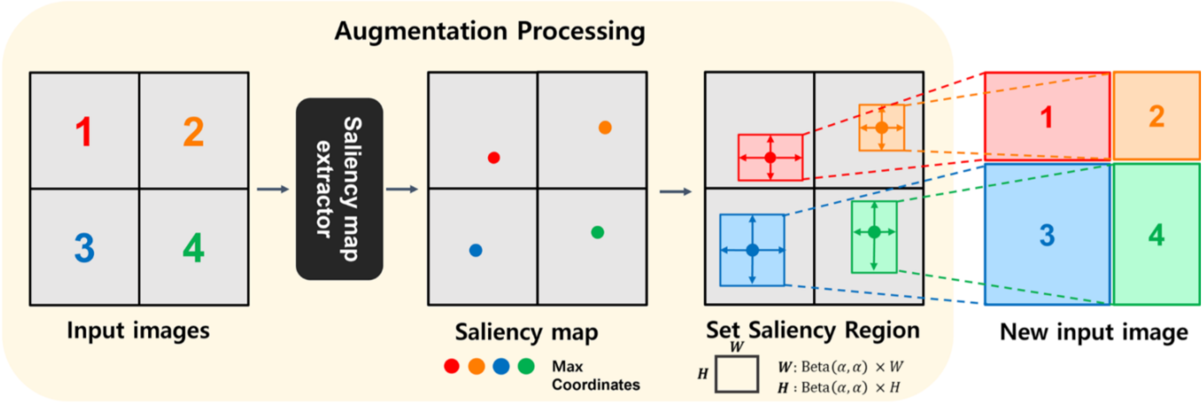


Figure 18: Confusion Matrix of VGGNet19, ResNet50, DenseNet121, MobileNetV2 and EfficientNetB0 for Saleicy Gradient Augmentation

Evaluation of the various convolutional neural network (CNN) models reveals intriguing insights concerning their relative performance. The results acquired by VGGNet are encouraging; they show a consistent upward trend throughout both the training and validation stages. ResNet50, meanwhile, shows notable improvements as seen by a drastic drop in loss, indicating superior performance and enhanced learning. However, as training nears completion, there is a chance of mild overfitting arising due to the widening gap between training and validation accuracy. DenseNet121 presents an out-of-the-ordinary scenario since it increases accuracy as it progresses through the epochs while concurrently decreasing loss. Nonetheless, the validation curve suggests overfitting is possible, requiring cautious assessment. The first 50 epochs of MobileNetV2 show promising behaviour, with accuracy increasing and loss decreasing. The conduct levels out after around 30 epochs, which is probably when it's at its most efficient. Overfitness may be the result of excessive training, therefore it's important to strike a balance. While the loss and accuracy of EfficientNetB0 both increase during the course of training, the validation accuracy still lags behind that of the training accuracy. Attempts to adjust the learning rate based on metrics that are currently unavailable demonstrate the complexity of fine-tuning. It becomes clear that Saliency Augmentation techniques are essential to enhancing model performance generally. DenseNet's massive performance boost is a shining example of this. Specific class labelling, such as "deer" and "cat," and the positive effect of Saliency Augmentation, further illustrate the intricate relationship between model construction and data augmentation techniques.

Discussion



. **Figure 19:** A framework depicting Saliency Augmentation

Table 1: Overall Performance Comparison using CIFAR10 dataset

Model Name	Baseline			Random Erasing Augmentation			Saliency graph Augmentation		
	Accuracy	Epoch	Loss_Epoch	Accuracy	Epoch	Loss_Epoch	Accuracy	Epoch	Loss_Epoch
VGG19	**0.9927**	47	0.0198	0.9929	45	0.218	0.9918	50	0.0268
ResNet50	0.9466	44	0.1751	0.9672	50	**0.1095**	**0.925**	50	0.144
DenseNet121	0.7649	43	0.7177	0.1003	47	2.3027	**0.9703**	48	0.0916
MobileNetV2	0.7433	48	0.7764	0.7431	47	0.7702	**0.9023**	50	0.2895
EffiecientNetB0	0.8148	43	0.5702	0.8305	50	0.5243	**0.9345**	49	0.2434

Saliency Augmentation techniques play a pivotal role in enhancing the overall performance of various convolutional neural network (CNN) models. Their impact is particularly pronounced in DenseNet, where a significant improvement in performance is observed. The augmentation technique not only aids in improving classification accuracy but also contributes to refining the models' ability to generalize and handle complex patterns within the data. However, while Saliency Augmentation proves effective in most cases, it is important to note that its influence varies across different models. In some instances, such as MobileNetV2, the technique helps achieve a balanced and optimal performance plateau, preventing overfitting and ensuring steady learning. This underscores the value of Saliency Augmentation in fine-tuning model behavior and addressing challenges associated with classification tasks. Overall, the integration of Saliency Augmentation strategies emerges as a promising avenue for boosting the capabilities of CNN models, with the potential to significantly improve accuracy, generalization, and the models' overall capacity to handle diverse and complex datasets.

In order to check the validation of the models performance, we selected FashionMNIST dataset to test upon. The results is as below,

Table 2: Fashion MNIST data performance comparison

Data	Normal			Random Erasing			Saliency		
	Accuracy	Epoch	Loss	Accuracy	Epoch	Loss	Accuracy	Epoch	Loss
VGG19	0.9228	50	0.227	0.88	47	0.3165	**0.9486**	39	0.1341
ResNet50	0.9178	50	0.3016	0.9108	45	0.2167	**0.923**	50	0.1408

We can see, like the performance of VGGNet in above scenarios, in this case it has outperformed in case of the saliency augmentation. The reason might be the images are gray scaled and the information extracted through the gradient is corelating with the classes labels. Saliency worked better on VGG19 whereas it as almost closely similar to performance with the ResNet50. There is not major difference in the accuracy value. Accuracy terms are quiet appreciable.

7 Conclusion

The exceptional potential of augmentation tactics has been demonstrated via their study and use, with the Saliency Augmentation methodology standing out as the most promising. Models like VGGNet, ResNet50, DenseNet121, MobileNetV2, and EfficientNetB0 showed varying degrees of sensitivity to various treatments. In particular, Saliency Augmentation proves to be a valuable tool for reducing overfitting, improving generalisation, and boosting classification precision. Although the vast majority of models benefit from Saliency Augmentation, its impact differs depending on the specifics of each one. It integrates smoothly with DenseNet121, yielding a significant performance gain. However, the approach is useful in certain cases, such as MobileNetV2, for obtaining a balanced and optimal performance plateau. This highlights the significance of adapting augmentation methods to the unique topologies of various model classes.

Saliency Augmentation has proven to be effective, and this has broader implications for how we approach problems in computer vision and deep learning. The effective implementation of Saliency Augmentation demonstrated this potential. It's obvious that augmentation tactics like saliency augmentation will play a crucial role in the ongoing effort to increase the capabilities of AI systems. By effectively addressing issues including insufficient data, overfitting, and generalisation, these methods pave the way for the creation of models that are more robust, accurate, and adaptable. Using AI to its fullest potential as a helper in human efforts requires unique augmentation tactics, which have recently been included. This study emphasises the importance of further investigation and experimentation with augmentation strategies for the purpose of gaining new insights and approaches that can propel deep learning to even greater heights. We are getting closer to reaching the full potential of AI in terms of tackling challenging challenges in the real world thanks to efforts like these and the continual advancement of technology.

8. References

Adhikari, N.D., 2020. Infection severity detection of CoVID19 from X-Rays and CT scans using artificial intelligence. International Journal of Computer (IJC), 38(1), pp.73-92.

An, J., Jang, S., Kwon, J., Jin, K. and Kim, Y., 2022. Saliency guided data augmentation strategy for maximally utilizing an object's visual information. Plos one, 17(10), p.e0274767

Boyd, A., Bowyer, K.W. and Czajka, A., 2022. Human-aided saliency maps improve the generalization of deep learning. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (pp. 2735-2744)]

Cheng, M.M., Hou, Q.B., Zhang, S.H. and Rosin, P.L., 2017. Intelligent visual media processing: When graphics meets vision. Journal of Computer Science and Technology, 32, pp.110-121

Cong, R., Lei, J., Fu, H., Cheng, M.M., Lin, W. and Huang, Q., 2018. Review of visual saliency detection with comprehensive information. IEEE Transactions on circuits and Systems for Video Technology, 29(10), pp.2941-2959

Ferreira, C.A., Melo, T., Sousa, P., Meyer, M.I., Shakibapour, E., Costa, P. and Campilho, A., 2018, June. Classification of breast cancer histology images through transfer learning using a pre-trained inception resnet v2. In International conference image analysis and recognition (pp. 763-770). Cham: Springer International Publishing

Fuhl, W., 2022. Technical report: Combining knowledge from transfer learning during training and wide resnets. arXiv preprint arXiv:2206.09697

Ghose, D., Desai, S.M., Bhattacharya, S., Chakraborty, D., Fiterau, M. and Rahman, T., 2019. Pedestrian detection in thermal images using saliency maps. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0)

Gorji, S. and Clark, J.J., 2017. Attentional push: A deep convolutional network for augmenting image salience with shared attention modeling in social scenes. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 2510-2519)

Gorji, S. and Clark, J.J., 2018. Going from image to video saliency: Augmenting image salience with dynamic attentional push. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 7501-7511)

Gu, K., Zhai, G., Yang, X., Zhang, W. and Chen, C.W., 2014. Automatic contrast enhancement technology with saliency preservation. IEEE Transactions on Circuits and Systems for Video Technology, 25(9), pp.1480-1494

Kucerova, J. and Sikudova, E., 2011. Saliency map augmentation with facial detection. In Proceedings of the 15th Central European seminar on computer graphics

Kuen, J., Wang, Z. and Wang, G., 2016. Recurrent attentional networks for saliency detection. In Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (pp. 3668-3677)]

Kumar, T., Brennan, R. and Bendechache, M. (2022) “Stride Random Erasing Augmentation,” CS & IT Conference Proceedings, 12(2). Available at: https://csitcp.com/paper/12/122csit01.pdf.

Lévy, D. and Jain, A., 2016. Breast mass classification from mammograms using deep convolutional neural networks. arXiv preprint arXiv:1612.00542

Mathe, S. and Sminchisescu, C., 2012. Dynamic eye movement datasets and learnt saliency models for visual action recognition. In Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part II 12 (pp. 842-856). Springer Berlin Heidelberg

Mechrez, R., Shechtman, E. and Zelnik-Manor, L., 2019. Saliency driven image manipulation. Machine Vision and Applications, 30(2), pp.189-202

Naushad, R., Kaur, T. and Ghaderpour, E., 2021. Deep transfer learning for land use and land cover classification: A comparative study. Sensors, 21(23), p.8083

Obeso, A.M., Benois-Pineau, J., Vázquez, M.G. and Acosta, A.R., 2019. Saliency-based selection of visual content for deep convolutional neural networks: Application to architectural style classification. Multimedia Tools and Applications, 78, pp.9553-9576

Sandor, C., Cunningham, A., Dey, A. and Mattila, V.V., 2010, October. An augmented reality x-ray system based on visual saliency. In 2010 IEEE international symposium on mixed and augmented reality (pp. 27-36). IEEE

Shorten, C. and Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. Journal of big data, 6(1), pp.1-48

Walambe, R., Marathe, A. and Kotecha, K., 2021. Multiscale object detection from drone imagery using ensemble transfer learning. Drones, 5(3), p.66.

Zhang, J., Liu, Y., Zhang, S., Poppe, R. and Wang, M., 2020. Light field saliency detection with deep convolutional networks. IEEE Transactions on Image Processing, 29, pp.4421-4434

Zhang, Z., Xu, Y., Yu, J. and Gao, S., 2018. Saliency detection in 360 videos. In Proceedings of the European conference on computer vision (ECCV) (pp. 488-503)

Zhao, Y., Ma, J., Li, X. and Zhang, J., 2018. Saliency detection and deep learning-based wildfire identification in UAV imagery. Sensors, 18(3), p.712

Zhu, C., Li, G., Wang, W. and Wang, R., 2017. An innovative salient object detection using center-dark channel prior. In Proceedings of the IEEE international conference on computer vision workshops (pp. 1509-1515)