COMP534 – APPLIED ARTIFICIAL INTELLIGENCE

ASSIGNMENT - 3

Solving Image Classification Problems with Convolutional Neural Networks

Jishnu Prakash Kunnanath Poduvattil
201581347
j.kunnanath-poduvattil@liverpool.ac.uk

Akhil Raj
201594703
a.raj@liverpool.ac.uk

Introduction

This report consists of comparison between VGGNet and ResNet, discussion and implementation of new models that can improve the performance of the best pre-trained model used for image classification.

Libraries Used

We mainly used the following libraries for modelling and Exploratory data analysis (EDA): -

Library	Utilisation	Link
Torch	Build, train, and test CNN models	https://pytorch.org/
Torchvision	Load pretrained models	https://pytorch.org/
Numpy	Numerical operations	https://numpy.org/
Scikit-learn	Classification Report and Confusion Matrix	https://scikit-learn.org/stable/
Pandas	Dataframe from dictionary	https://pandas.pydata.org/
Matplotlib	Plotting diagrams	https://matplotlib.org/
Seaborn	charts for image distribution	https://seaborn.pydata.org/

Data Cleaning and Analysis Process

The given image classes are imbalanced as shown in the above plot. We have 3418 images for Pneumonia, 1266 for Normal and 460 images for Covid19 classes. To train the models with imbalanced data, class weights can be initialised with loss function [1]. Weights for each class is simply the number of images in a class divided by the total number of training images.

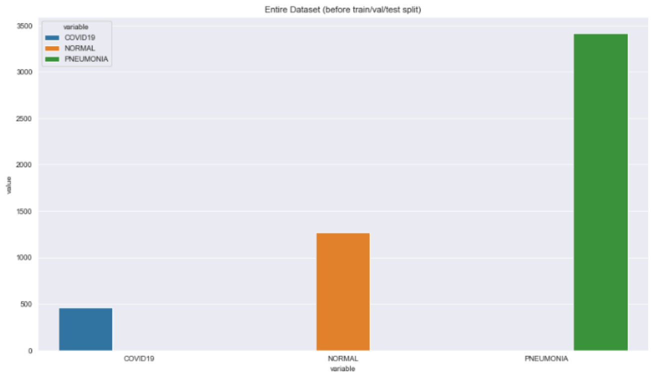


Figure 1: No. of Images per class

Data Augmentation and pre-processing

The images were augmented and pre-processed and resized to (224,224) size for training and testing. The techniques used are discussed below.

Train Images

Augmentation/Pre-processing	Description
Resize	Resize image to a specific size (256, 256)
Random Resized Crop	Random crops of image and resize to (224, 224)
Random Horizontal Flip	Flip the image horizontally with a probability of 50%.
To Tensor	Converts the image to tensor
Normalise	Normalise the values to specific mean and standard deviation

For test images, resize to (256, 256), Centre crop to (224,224), convert to tensor and normalisation was used.

Pre-Trained CNN models

We selected VGGNet and ResNet for the comparison of pre-trained CNN models.

1. **VGGNet**

VGGNet consists of 16 convolutional layers, and it has a very uniform architecture. Like AlexNet, only 3x3 convolutions, but lots of filters. Trained on 4 GPUs for 2–3 weeks. The weight configuration of the VGGNet is publicly available and has been used in many other applications and challenges as a baseline feature extractor. However, VGGNet consists of 138 million parameters, which can be a bit challenging to handle. Runner-up in ILSVRC Challenge 2014.

The simple architecture of VGG is the main reason for selecting this model. We selected a VGG-16 model with 16 weight layers that have combinations of 2 or 3 convolutional layers followed by a max pooling layer. Classification layers consist of 3 fully connected layers.

2. **ResNet**

Residual Neural Network (ResNet) introduced an architecture with “skip connections” and features heavy batch normalization. Such skip connections are also known as gated units or gated recurrent units. Thanks to this technique authors of [2] were able to train a neural network with 152 layers while still having lower complexity than VGGNet. Winner of ILSVRC Challenge 2015.

The concept of skip connections inspired us to select the Resnet-18 model. It consists of basic blocks of CBRCB combinations, which is a convolution layer- batch normalisation layer- ReLu activation layer-convolution layer- batch normalisation layer. 18 weight layers are present in the model with one fully connected layer at the end for classification. Also, according to the authors in the paper [3], the Resnet model gives the best performance.

Training and Testing Process

Train-Validation split of 80-20% was used for model training. SubsetRandomSampler module from torch was used to ensure random samples in train and validation sets. Learning rate scheduler is initialised in the training process. After every 7 epochs, the learning rate will decay by 0.1.

Evaluation

Performance Analysis of The Pre-Trained Models

1) **VGGNet**

We Loaded VGG-16 pretrained model with weights trained on ImageNet dataset of 1000 classes. We changed the final layer form 1000 classes to 3 classes and predict our test data. No training is done as suggested in the assignment specification. Precision, recall, f1-score, and accuracy are calculated for comparison. The model with the pre-trained weights gives 22% accuracy.

Class	precision	recall	f1-score
COVID19	0.09	0.69	0.17
NORMAL	0.45	0.63	0.53
PNEUMONIA	0.00	0.00	0.00

This model has very low accuracy and completely unable to classify at least one in pneumonia correctly. Covid19 has very low precision (when it predicts covid19, it is correct only 9% of the time) but has high recall (it correctly identifies 69% of all covid19). Prediction of normal X-rays has 0.53 f1-score.

2) **ResNet**

We Loaded ResNet-18 pretrained model. Similar to VGG-16, ResNet-18 also loaded with weights trained on ImageNet dataset of 1000 classes. We changed the final layer form 1000 classes to 3 classes and predict our test data. No training is done as suggested in the assignment specification. Precision, recall, f1-score, and accuracy are calculated for comparison. The model with the pre-trained weights gives 43% accuracy.

Class	precision	recall	f1-score
COVID19	0.10	0.27	0.15
NORMAL	0.28	0.35	0.31
PNEUMONIA	0.70	0.48	0.57

The model has an accuracy of 43% and has performed well in pneumonia class. The model has less performance in predicting normal and covid19 x-rays. Of the Covid19 prediction only 10% is correct and correctly identifies 27% of all covid19.

As per the results discussed above, ResNet performs better with our dataset with 43% accuracy compared to the 22% accuracy of VGGNet. ResNet also predicted all classes compared to VGGNet not predicting pneumonia class. Overall ResNet is better than VGGNet for our application.

New Proposed Networks

We propose two networks: (1) fine tuning final layer in the resnet-18. (2) Adding C-B-R block with dropout

1) **Model 1 (Fine tuning final layer in the ResNet-18):** We train the final layer in the resnet-18 model that is fine tuned to predict 3 classes. Then all the layers were freezed for keeping the weights from ImageNet and the final layer was unfreezed for training. By initialising the training loss function with computed weights for each class, the data imbalance did not affect the model performance as expected. Stochastic gradient descent with a momentum of 0.9 is used as an optimiser with a learning rate of 0.001. These hyperparameters were selected for replicating the same hyperparameters used by the authors of [2]. The learning rate was selected based on intuition and by observing several articles, videos etc.
2) **Model 2 (Adding C-B-R block with dropout):** One of the specialities of Resnet is the usage of C-B-R combinations in its basic block. The combination of convolution layer followed by batch normalisation and ReLu activation layer is added before average pooling with a dropout node of 50% probability. For classification, 2 layers are added for predicting 3 classes. The model is trained, and the weights are updated for all layers. The hyperparameters remain the same as above.

Results (New Networks)

Model 1 (Fine tuning final layer in the ResNet-18):

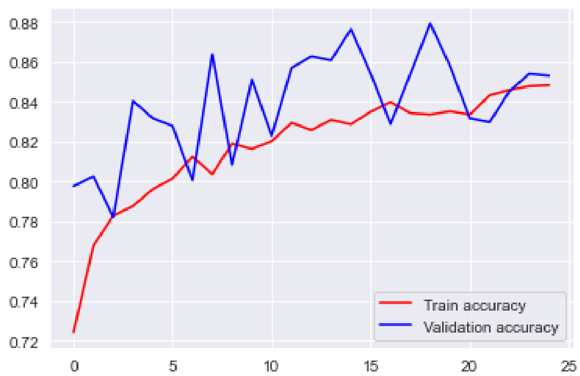


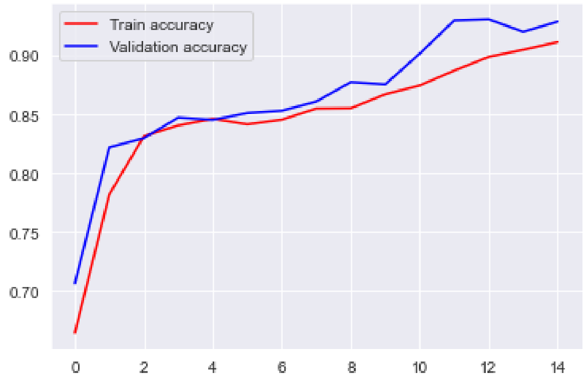
Figure 2.1: Train and Validation accuracy

2.2: Train and Validation loss

Class	precision	recall	f1-score
COVID19	0.97	0.75	0.84
NORMAL	0.92	0.81	0.87
PNEUMONIA	0.91	0.98	0.94

We can observe good performance here with an accuracy of 92%. The model correctly identifies 75% of all covid19 which is the lowest recall score for any class. By simply training the final layer, we observed substantial improvement in model performance. But a slight instability can be seen in the evaluation graphs. Another iteration of training was conducted with a learning rate scheduler with above specified parameters, but the performance was not satisfactory. Although, the minor variation in validation did not affect the model performance on the test data.

Model 2 (Adding C-B-R block with dropout):



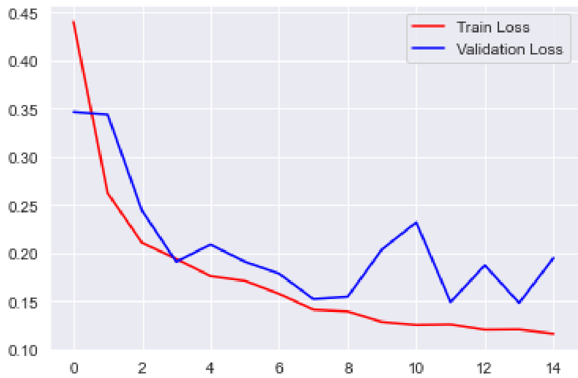


Figure 3.1: Train and Validation accuracy

3.2: Train and Validation loss

Class	precision	recall	f1-score
COVID19	1.00	0.91	0.95
NORMAL	0.92	0.90	0.91
PNEUMONIA	0.96	0.98	0.97

The model gives best performance of all classes with an overall test accuracy of 95%. By adding C-B-R block, an adequate improvement in the model performance can be observed. The final C-B-R block was the best performing combination in many different training iterations conducted. Different combinations of Original Residual block of C-B-R-C-B (convolution- batch normalisation- ReLu activation - convolution- batch normalisation) with a max pooling layer and dropout were considered and trained. Training the whole model with weights initialised from ImageNet definitely gave fast convergence. The high recall values for covid19 and pneumonia suggests that the model gives high importance in detecting the infections.

Comparison

The model 2 performed extremely well compared to the pretrained models with the pretrained weights. Since the pretrained models are trained in the ImageNet database, the prediction for medical data with the weights is not accurate. After training only the final layer of the pretrained ResNet-18 model in model 1, the accuracy is significantly improved (from 43% to 92%). By adding C-B-R block with dropout in ResNet-18 (model 2) and training, we were able to achieve an accuracy of 95%.

Model	Train Accuracy	Validation Accuracy	Test Accuracy
Model 1	85	85	92
Model 2	91	93	95

Future Work

1. Find updated image datasets for covid19 with more quality, and train on these models to compare performance.
2. There are many aspects that make transfer learning interesting and challenging. There are other models with more layers and complex architectures that performed well in the ImageNet dataset. With enough time and computational power, we can try to train those models on our image data.
3. Training model with random weight initialization instead of using ImageNet weights for comparing model performances.

Final Conclusions

Challenges

The main challenge of this project was working with Imbalance dataset. We tried two methods to resolve this issue.

1. **Image Augmentation**
   The covid19 class only had 460 images. So, using image augmentation described in the data analysis section and image rotation, we created 1500 images for covid19. Then selected 1500 images randomly from Pneumonia and Normal image classes and used it for training. The model performance was comparatively not satisfactory.
2. **Computing Weights for image class**
   Weights were computed for each class (reference and details in data analysis section) and initialised with loss function.

Apparently the second method outperformed the first. We think it is because of poor data distribution.

Task Allocation

Jishnu Prakash Kunnanath Poduvattil

Research and development for model selection, training and testing using pytorch. Data augmentation and pre-processing and Documentation.

Akhil Raj

Research and documentation for the pretrained models. Prepared report and compared the results of each model.

References

[1] Cui, Yin, et al. “Class-Balanced Loss Based on Effective Number of Samples.” 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019. Crossref, https://doi.org/10.1109/cvpr.2019.00949.

[2] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.

[3] Raghu, Maithra & Zhang, Chiyuan & Kleinberg, Jon & Bengio, Samy. (2019). Transfusion: Understanding Transfer Learning with Applications to Medical Imaging.