Weekly Report

Comparative Analysis of Image Classification Models for Efficient and Accurate Classification across Diverse Image Types

Student: Yusif Mukhtarov

Date: 23/07/2023

Short info about Research

The fundamental objective of this **quantitative** research is to identify the best image classification models in terms of both time efficiency and accuracy, for images of low quality. Several widely recognized models, including AlexNet, VGGNet, ResNet, EfficientNet will be considered for this analysis. **Qualitative** analysis has been made to pick those models since they are considered by experts to be the best.

The dataset chosen for this experiment will be vegetable custom datasets that incorporate the required image conditions. Data preprocessing will be carried out to facilitate the suitable condition for each model, involving adding noise.

To achieve an in-depth and unbiased analysis, the study will take into account several performance metrics. Classification accuracy will primarily measure the model's performance. Simultaneously, precision, recall, and F1-Score will also be considered to provide a good understanding of the model's performance. Moreover, the time taken for training and prediction will be recorded to measure the model's efficiency. Additionally, the model's size in terms of parameters will also be observed.

Have been done:

This week I was finally able to train my models. So, the whole week I was trying out different parameters for training, (schedulers, momentum etc) to reach good accuracy. I trained all 4 models: AlexNet, EfficientNet, Vgg16 and ResNet. All of the training outputs are available in my github in training_in_kaggle.ipynb file. These are the results of testing those models on images with 10% noise.

```
[15]
1  alexnet = torch.load('../models/alexnet_without_noise.pth', map_location=torch.device('cpu'))
2  testing(alexnet)
```

```
Accuracy of the network: 96.13333333333334 %
Accuracy of Bean: 96.5 %
Accuracy of Bitter_Gourd: 99.0 %
Accuracy of Bottle_Gourd: 96.0 %
Accuracy of Brinjal: 89.0 %
Accuracy of Broccoli: 99.5 %
Accuracy of Cabbage: 99.0 %
Accuracy of Capsicum: 74.5 %
Accuracy of Carrot: 99.5 %
Accuracy of Cauliflower: 98.0 %
Accuracy of Cucumber: 98.5 %
```

```
[16]
1  resnet = torch.load('../models/resnet_without_noise.pth', map_location=torch.device('cpu'))
2  testing(resnet)
```

```
Accuracy of the network: 99.2 %
Accuracy of Bean: 100.0 %
Accuracy of Bitter_Gourd: 99.0 %
Accuracy of Bottle_Gourd: 100.0 %
Accuracy of Brinjal: 99.5 %
Accuracy of Broccoli: 98.5 %
Accuracy of Cabbage: 99.5 %
Accuracy of Capsicum: 99.5 %
Accuracy of Carrot: 98.5 %
Accuracy of Cauliflower: 98.0 %
Accuracy of Cucumber: 98.0 %
```

```
[21]
efficientnet = torch.load('../models/efficientnet_without_noise.pth', map_location=torch.device('cpu'))
testing(efficientnet)
```

```
Accuracy of the network: 96.53333333333333 %
Accuracy of Bean: 99.5 %
Accuracy of Bitter_Gourd: 99.0 %
Accuracy of Bottle_Gourd: 87.0 %
Accuracy of Brinjal: 98.0 %
Accuracy of Broccoli: 97.0 %
Accuracy of Cabbage: 98.0 %
Accuracy of Capsicum: 97.0 %
Accuracy of Carrot: 97.5 %
Accuracy of Cauliflower: 95.5 %
Accuracy of Cucumber: 96.0 %
```

```
[22]
vgg16 = torch.load('../models/vgg16_without_noise.pth', map_location=torch.device('cpu'))
testing(vgg16)
```

```
Accuracy of the network: 96.3 %
Accuracy of Bean: 100.0 %
Accuracy of Bitter_Gourd: 99.0 %
Accuracy of Bottle_Gourd: 87.0 %
Accuracy of Brinjal: 93.5 %
Accuracy of Broccoli: 99.5 %
Accuracy of Cabbage: 99.0 %
Accuracy of Capsicum: 79.5 %
Accuracy of Carrot: 98.5 %
Accuracy of Cauliflower: 95.0 %
Accuracy of Cucumber: 99.0 %
```

I finished all training processes just yesterday, so I haven't created a good table for those results yet (I will get such results for images with 30%, 50%, 70% and 100% noise). But I am planning to create a lot of charts, graphs and tables to get a lot of insights and outcomes for the final report and presentation. For next week I am planning to have a meeting with Professor Kaisler to discuss what I have accomplished so far and what I can do more to achieve better results. However, I have my own plans as well. I want to integrate WanDB into my training algorithms to get good graphs, tables and illustrative charts of the training processes from which again a lot of insights and outcomes can be gotten since Weights & Biases (WandB) is a machine learning development platform that allows users to track and visualize various aspects of their model training process in real-time.