Student: Chingis Oinar
Kaggle Team Name: RussianFox
Score: 0.96333
Student ID: 2018315169

Final Project Report

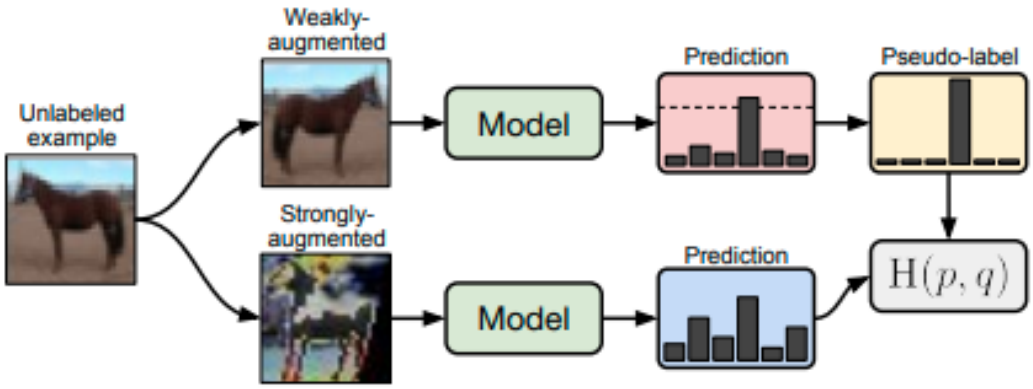

Figure 1: Diagram of FixMatch

Method

To tackle the given problem I used the approach proposed in the paper titled *"FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence"* [1]. I also tried the approach proposed in the paper titled "*Unsupervised Data Augmentation for Consistency Training*", however; Fixmatch gave me better results [2]. The training process is shown in the figure attached above. As it is seen, a single example x is augmented using both weak and strong augmentations. Subsequently, a model makes predictions for both images, however; the weakly augmented sample is used in order to produce a pseudo label. As it can be observed, there is a certain threshold which is used to make sure a model is not trained on the uncertain samples. Thus, the model is trained to make its prediction on the strongly-augmented version match the pseudo-label via a cross-entropy loss. Additionally, labeled samples are utilized to perform the conventional supervised approach.

Augmentations

The authors utilize RandAugment and CTAugment to produce strongly-augmented samples, however; I only used RandAugment for the given problem. Additionally, I used Random Horizontal Flip and Random Rotation to produce weakly-augmented samples. There are two hyperparameters for RandAugment that are the number of augmentations applied sequentially and the magnitude for all transforms. I set them to 2 and 10 respectively.

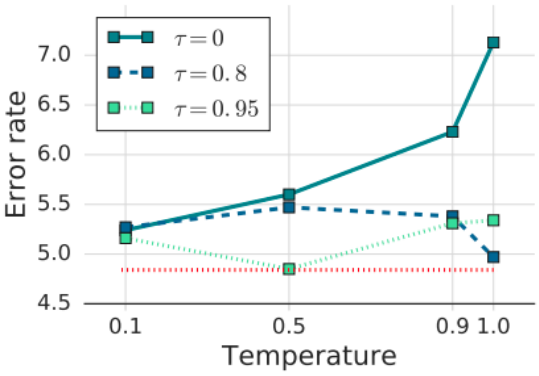

Figure 2: Error rate against temperature for three different threshold values.

Hyperparameters and Settings

Most predictions have low confidence compared to the labels we use. So,
sharpening the pseudo-label may enhance the performance. The hyperparameter for that is introduced as a temperature. The figure above shows some results obtained by the authors of the paper. Thus, I used 0.5 for temperature and 0.95 for threshold. The learning rate was set to 1e-4 and the batch size is 64. The labeled train set was split into train and validations subsets, where the ratio is 0.1. All the models were trained for 100 epochs. Finally, the images are resized into 128 by 128, which gave me a better performance. Finally, all the networks used were pretrained on the ImageNet dataset.

Experimental Results

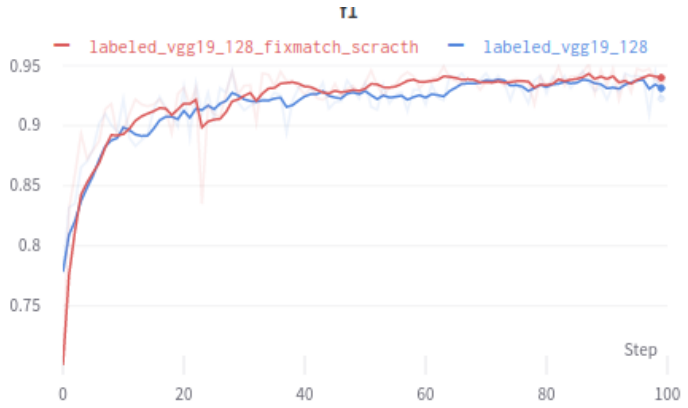

Figure 3: F1 score for VGG 19.

The figure above shows the performance on the validation subset of VGG19 network trained using both Fixmatch and only labeled data. Although their performances on the validation subset are similar, Fixmatch gave me a better result on Kaggle.
The following figure depicts performances for five different architectures, which are ResNet 18, VGG19 with Batch Normalization, MobileNet v2, VGG 16 and VGG 19.

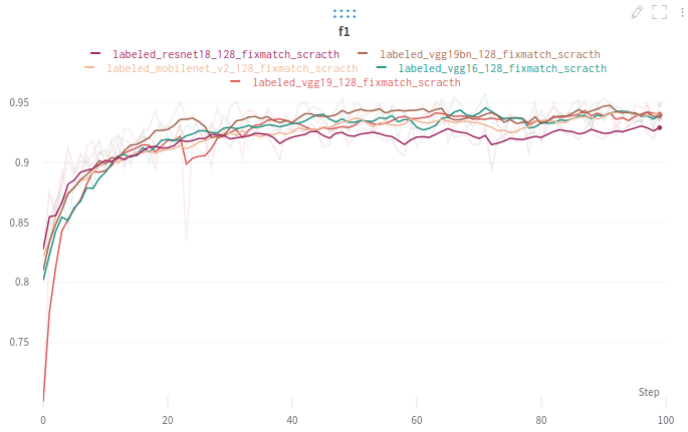

Figure 4: F1 score for ResNet 18, VGG19 with Batch Normalization, MobileNet v2, VGG 16 and VGG 19.

As it is seen their performances are approximately similar on the validation subset, however; VGG 19 with Batch Normalization performed slightly better.

Generally, VGG 19 architectures performed better on Kaggle as well.
Finally, the last figure shows their validation and train losses over 100 epochs, where dashed lines are for train losses.

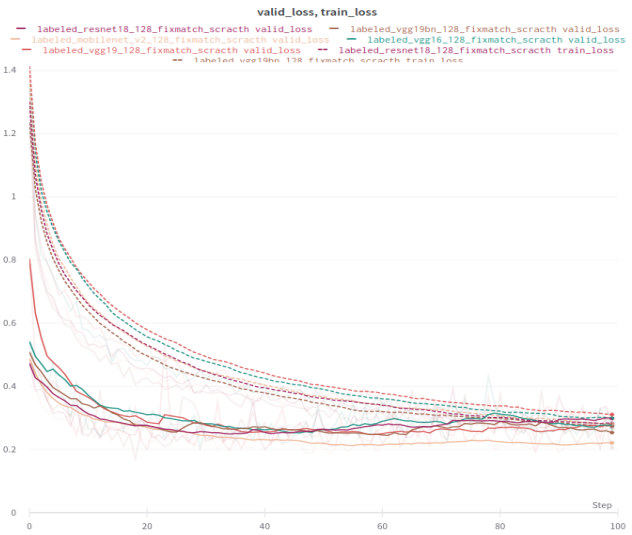

Figure 5: losses for ResNet 18, VGG19 with Batch Normalization, MobileNet v2, VGG 16 and VGG 19.

Kaggle Results

Network	Accuracy
VGG 16	0.92966
VGG 19	0.93600
ResNet 18	0.92
MobileNet v2	0.92633
VGG 19 BN	0.93566

Table 1: Accuracy scores.

As it is seen VGG 19 gave better results on the Test set, which gave an accuracy score of 0.93600. Overall, the VGG family of networks worked better on the dataset. However, my final score is 0.94366. So, I utilized the Ensemble Learning Method in order to improve the performance overall. Thus, by aggregating several architectures I managed to improve on my result. Finally, I utilized a soft voting method during inference.

Ensemble	Accuracy
VGG16,VGG19,MobileNet v2	0.94
VGG16,VGG19,MobileNet v2, VGG19 BN	0.94366
VGG16,VGG19,MobileNet v2, VGG19 BN, ResNet 18	0.94333

Table 2: Accuracy scores for Ensembles.

Further Improvements

I increased the image resolution up to 224. Thus, the following results are the improvements after this change.

Network	Accuracy
VGG19, VGG16, VGG19_bn,MobilenetV2	0.95966
ResNet18, VGG19, VGG16	0.95800
ResNet18, VGG19, VGG16, VGG19_bn	0.96066
ResNet18, VGG19, VGG16, VGG19_bn,MobilenetV2	0.96333
ResNet50, VGG19, VGG16, VGG19_bn,MobilenetV2	0.96433

References

[1]Sohn, K., “FixMatch: Simplifying Semi-Supervised Learning with Consistency and Confidence”, arXiv e-prints, 2020.
[2]Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V., “Unsupervised Data Augmentation for Consistency Training”, arXiv e-prints, 2019.