AAIT

Experimental Report

Task 1

The initial task involved tackling the challenge of classification in the presence of incomplete labeling. Specifically, the training dataset comprised approximately 50,000 samples, half of which lacked labels, with 100 different classes. Additionally, there was a separate test set containing 5,000 samples. To facilitate effective model evaluation and tuning, I partitioned the original dataset into two distinct subsets: a training set and a validation set. The division was executed with a specific allocation ratio, where 20% of the samples were reserved for validation purposes. Another thing to mention here was the fact that the split took into account the need to have equal amount of samples from each classes for both subsets.

In order to introduce variability in the dataset, without collecting new data, and also to increase its size I used Data Augmentation. This process consists of making minor alterations to the existing data, including flipping the image horizontally, vertically, cropping, rotating, scaling, etc. In my pipeline I included the following transformations: random horizontal flip and random rotation of 15 degrees. This technique also helps with overfitting.

The preliminary approach adopted was a baseline methodology. This method primarily focused on training exclusively on the portion of the training dataset that possessed labels, effectively disregarding the unlabeled segment of the training set. Post-training, this model was then employed to predict labels for the test set. In my quest to identify the most efficacious model for this task, I experimented with two different architectures: EfficientNetB0 and ResNet50. Through comparative analysis and performance evaluation, it became evident that the ResNet50 model demonstrated superior performance in this specific context, outperforming its EfficientNetB0 counterpart.

Model	Accuracy	Precision	Recall	F1 score
ResNet50	71.3%	71.8%	71.3%	71.4%
EfficientNetB0	57.8%	57.7%	57.3%	57.6%

Table 1: ResNet50 vs EfficientNetB0 after training on 30 epochs

In the process of fine-tuning the model parameters for optimal performance, I discovered that the choice of batch size and the type of optimizer used had significant impacts on the results. After experimenting with various batch sizes, it was determined that a batch size of 64 yielded the best outcomes. This specific batch size struck an effective balance, providing sufficient data in each iteration to enable stable and efficient learning, while also maintaining manageable computational requirements. Another critical factor that substantially influenced the model's performance was the selection of the optimizer. In this context, I compared the efficacy of different optimization algorithms, particularly focusing on Stochastic Gradient Descent (SGD) and Adam. Through comparative analysis, it became evident that Stochastic Gradient Descent held a distinct advantage over Adam in this specific scenario. SGD's performance was notably superior, leading to better convergence and more robust results compared to the outcomes when using the Adam optimizer. The results in the following table represent the comparison between Adam and SGD optimisers for the ResNet50 model, after training on 30 epochs.

Optimiser	Accuracy	Precision	Recall	F1 score
Adam	70%	70.3%	69.9%	70.1%
SGD	71.3%	71.8%	71.3%	71.4%

Table 2: ResNet50 Adam vs SGD performance

Building upon the foundational method, I ventured into implementing a more sophisticated technique, commonly referred to in the literature as the "Noisy Student" approach. This method is an iterative process that begins with the training of an initial model, often termed the 'teacher' model. The training of this teacher model is exclusively conducted on the subset of the training data that is labeled. Once the teacher model is adequately trained, it is then employed to generate what are known as 'pseudo labels' for the unlabeled portion of the training dataset. The next phase involves the introduction of a 'student' model. This student model undergoes training on a dataset that combines both the originally labeled data and the newly pseudo-labeled data created by the teacher model. The objective is to harness the broader and enriched dataset to enhance the learning and generalization capabilities of the student model. Subsequently, this student model is utilized to generate labels for the test set.

Given the insights and performance metrics obtained from the initial method, I selected the ResNet50 model as the foundation for both the teacher and student models. In the training phase of the teacher model, which spanned over 30 epochs, the model achieved a training accuracy of 97.22%. Furthermore, in the validation phase, it attained a 71.3% across all key classification metrics, including precision, recall, F1 score, and overall accuracy. Upon the successful training and validation of the teacher model, focus shifted to the student model. The student model was rigorously trained over a similar duration of 20 epochs, but this time on the combined dataset, comprising both originally labeled and pseudo-labeled data. This comprehensive training regimen enabled the student model to achieve a training accuracy of 95%.

To definitively assess whether the Noisy Student method enhanced model performance compared to the baseline approach, the ultimate test involved evaluating and comparing their accuracies on the test set. After meticulously training the models using both methodologies, I submitted their respective best results for evaluation against the test set. The Noisy Student method demonstrated a notable improvement over the baseline method. Specifically, the model trained using the Noisy Student approach achieved an accuracy of 72.3% on the test set. This was a significant enhancement when compared to the baseline model, which managed to secure an accuracy of 70.2%.

Task 2

The second task involved tackling the challenge of classification in the presence of noisy labels. The training dataset comprised approximately 50,000 samples with 100 distinct classes. Additionally, there was a separate test set containing 5,000 samples. For the purpose of conducting a thorough evaluation and fine-tuning of the model, the original dataset was divided into two separate segments: one designated for training and the other for validation. This segmentation was carried out with a deliberate proportioning strategy, wherein 20% of the total samples were allocated to the validation set. Additionally, it's important to highlight that this split was carefully orchestrated to ensure a balanced representation of each class in both the training and validation subsets. The data augmentation is the same with the one implemented for the first task. The batch size values was kept at 64.

In the initial phase of my experimentation, the baseline methodology involved the utilization of a ResNet50 model, which was trained comprehensively on the

entire training dataset. This training process spanned across 20 epochs and employed SGD as the optimization technique. Upon completion of the training phase, the model achieved an accuracy of 84% on the training set. When evaluated on the validation set, the model demonstrated a modest performance, with nearly all key classification metrics, such as precision, recall, and F1 score, reaching close to 57.3%.

Subsequent to establishing this baseline, my focus shifted towards exploring strategies that could potentially enhance these results further. One such strategy entailed the construction of an ensemble of models, wherein each model in the ensemble would contribute to determining the final label by means of a voting mechanism. For the ensemble, in addition to the previously utilized ResNet50, I opted to integrate two additional models: EfficientNetB0 and MobileNetV2. The selection of these models was strategic, as both EfficientNetB0 and MobileNetV2 are renowned for their similarities with ResNet50 in terms of overall performance and computational efficiency. The results on the validation set were better than the baseline model, accuracy, recall, precision and f1 score reaching close to 75%.

In order to conclusively determine if the ensemble approach offered an improvement in performance relative to the baseline method, the key determinant was a comparative analysis of their accuracies on the test set. This comparison served as the critical evaluation to gauge the efficacy of each method. Following a rigorous and detailed training process with both the ensemble and baseline methodologies, I proceeded to evaluate the best-performing results of each approach against the test set. The model that was developed and trained under the ensemble strategy achieved an accuracy rate of 71.6% on the test set. This result represented a substantial improvement, especially when viewed in contrast to the performance of the baseline model, which achieved a slightly lower accuracy rate of 68.1%.

Link to the GitHub repo: https://github.com/Cosmeeeeen/aait-hw2