Submission: ZB4171 Project Report
Project Title: Leukaemia Image Classification

Names:
Amanda Ho Shan Rui (A0187715U)
Chen Tianying, Tiana (A0169834N)

Module Code: ZB4171 Advanced Topics in Bioinformatics
Professor: Professor Greg Tucker-Kellogg
Date: 21 November 2021

1. Introduction

Acute Lymphoblastic Leukaemia (ALL) is a blood and bone marrow cancer caused by uncontrolled proliferation and differentiation of lymphoblasts (Terwilliger & Abdul-Hay, 2017). ALL is the most common type of cancer in children, accounting for ~25% of all paediatric cancers. It can progress quickly and be fatal within weeks or months when left untreated. ALL diagnosis involves the identification of abnormal lymphoblasts from microscopic images, typically performed by an expert pathologist trained to identify key morphological patterns found in abnormal lymphoid cells. However, lymphoid blasts appear morphologically similar to healthy blood cells and are not easily distinguishable under the microscope. The inability to identify ALL cells accurately often leads to inter-observer variability where diagnoses are irreproducible when carried out by multiple pathologists (van der Laak et al., 2021).

In recent years, Artificial Intelligence (AI) and Machine Learning (ML) have shown promising applications in biology and medicine, especially in the field of clinical diagnostics (Yu et al., 2018). Several studies have demonstrated the robustness of AI systems with improved reproducibility and reduced inter-observer variability (van der Laak et al., 2021). There were previous applications of ML in leukaemia such as in bone marrow aspirates, but most of these applications deal with the identification of different blood cell types (Chandradevan et al., 2020). An AI classification model built specifically to distinguish abnormal and healthy cells within the same blood cell type will be useful and clinically relevant.

In this project, we aim to develop an image-based classification model using deep-learning techniques to distinguish between leukaemia blasts and healthy cells in microscopic images. Specifically, a convolutional neural network (CNN) model, EfficientNet, was trained with microscopic cell images obtained from the 'Leukemia Classification' challenge in Kaggle. Multiple ML techniques including Noisy Student training and ensemble framework were implemented to further improve EfficientNet model performance on ALL identification in microscopic images. We found that EfficientNet coupled with Noisy Student training and ensemble framework gave the most suitable model for the ALL classification task, achieving a high accuracy of ~90%.

2. Methodology

Project Overview

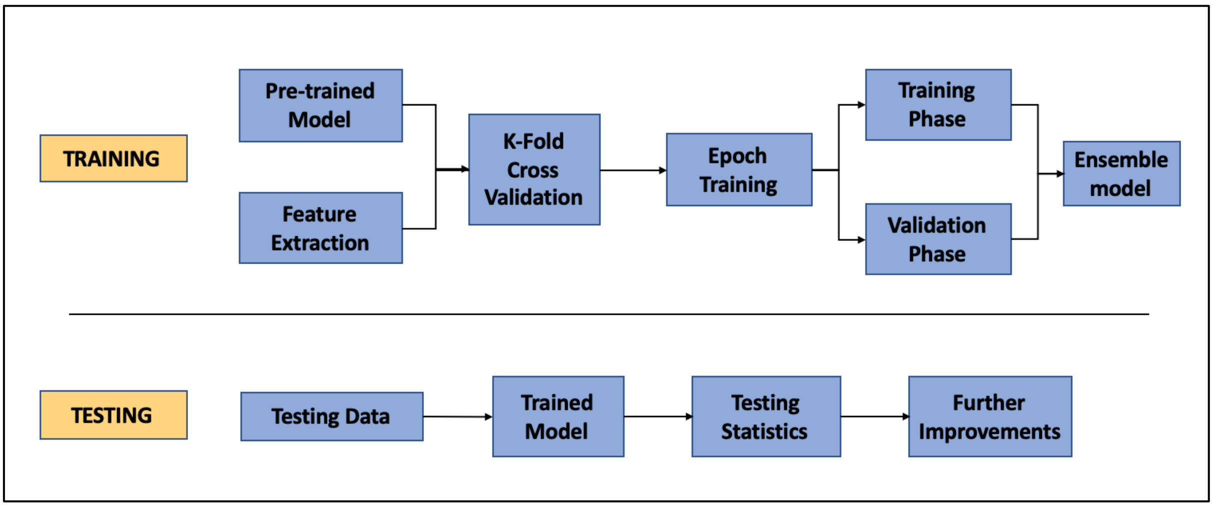


Figure 1 Overview of project methodology. A training and testing phase were carried out for each of the models implemented.

4 models were tested in this project including EfficientNet model, Ensemble EfficientNet model (EfficientNet and Ensemble framework), Noisy Student model (EfficientNet and Noisy Student training) and Ensemble Noisy Student model (EfficientNet, Noisy Student training and Ensemble framework). For each model, training and testing phases were carried out. In the training phase, pre-trained models were used to perform feature extraction on leukaemia images using k-fold cross validation with nested epoch training. Models generated from each of the k-folds were combined to form ensemble models. Ensemble models were tested and training parameters further adjusted for model improvement.

Dataset

The cell images used in this project is publicly available on Kaggle under the open challenge "Leukemia Classification" (Gupta et al., 2020). Training and validation sets were obtained from the challenge. Images are labelled by patient ID, image number and cell type ALL (leukaemia cell) or HEM (healthy blood cell). The testing set from the challenge was excluded because cell type labels of the testing images were not made publicly available. In total, 12,528 images from 101 patients were obtained for usage. The two sets were merged and split according to patient ID to obtain a final training and testing set for this project. 10,114 images from 86 patients were distributed to the training set and 2414 images from 15 patients were distributed to the testing set. The distribution of patients, images and cell type labels in the final training and testing sets are shown in Table 1.

Table 1 Distribution of patients, images and cell type labels in training and testing sets.

	HEM		**ALL**	
	No. of Patients	**No. of Images**	**No. of Patients**	**No. of Images**

Training Set	37	3459	49	6655
Testing Set	4	578	11	1836

EfficientNet Training

EfficientNet Architecture

An EfficientNet model was implemented for the ALL classification task. Transfer learning was applied to reduce training costs by initializing an EfficientNet-B0 model pre-trained on ImageNet (O et al., 2015). The pre-trained EfficientNet-B0 model is publicly available as a PyTorch Package on Github via https://github.com/lukemelas/EfficientNet-PyTorch. The classification layer of pre-trained EfficientNet model was changed to a binary output layer and yields a size of 1280. Feature extraction and training was then carried out on the pre-trained model using the training set.

EfficientNet Training

Stratified k-fold cross validation with k = 5 was carried out for EfficientNet model training. K = 5 was chosen due to limited training images obtained for usage from the Kaggle challenge. For each fold, the training set was partitioned into training and validation splits at a 4:1 ratio using a stratified sampling approach. Stratified sampling ensured that the ALL to HEM ratio were consistent with the non-partitioned training set in each fold. A Stochastic Gradient Descent (SGD) optimizer was used to minimize cross-entropy loss with an initial learning rate of $1e^{-3}$ and a batch size of 15. The learning rate was decayed by a factor of 0.1 every 7th epoch and training terminated at the 50th epoch. Each training fold produced 1 trained EfficientNet model and generated a total of 5 EfficientNet models from 5 folds.

Noisy Student Training

Noisy Student training was carried out on EfficientNet to improve model performance. We refer to the EfficientNet models trained used Noisy Student as the Noisy Student models. The training set was split into $(x, y)_{labelled}$, $(x, -)_{unlabelled}$ and $(x, y)_{validation}$ sets at a 2:2:1 ratio for Noisy Student training, with the cell-type labels of the unlabelled set masked. The same training parameters used for EfficientNet training was used for Noisy Student training. A new pre-trained EfficientNet-B0 model was initialised as the teacher model for every 1st Noisy Student iteration, and as the student model for all Noisy Student iterations.

3 Noisy Student iterations are carried out per training fold. In the 1st Noisy Student iteration, the teacher model was trained on $(x, y)_{labelled}$ for 50 epochs and used for prediction on $(x, -)_{unlabelled}$ to generate $(x, i)_{pseudo-labelled}$ where i refers to the pseudo-labels generated by the teacher model. The student model is then trained on $(x, i)_{pseudo-labelled}$ for 50 epochs. For subsequent iterations, the trained student model is initialised as the new teacher model and a new student model is initialised. Each training fold produced 1 trained Noisy Student model, generating a total of 5 Noisy Student models from 5 folds.

Training Ensemble Models

The 5 base EfficientNet models and 5 base Noisy Student models obtained from training were combined separately to form an Ensemble EfficientNet model and an Ensemble Noisy Student model. The ensemble models use majority voting to aggregate predictions of base models to obtain a final class prediction for an image. Majority voting outputs the base prediction that has the most votes as the final prediction for each image (Figure 2). The majority voting system is suitable for our ensemble classification task due to the odd number of base models for each ensemble, making tie-breaking events redundant during prediction voting.

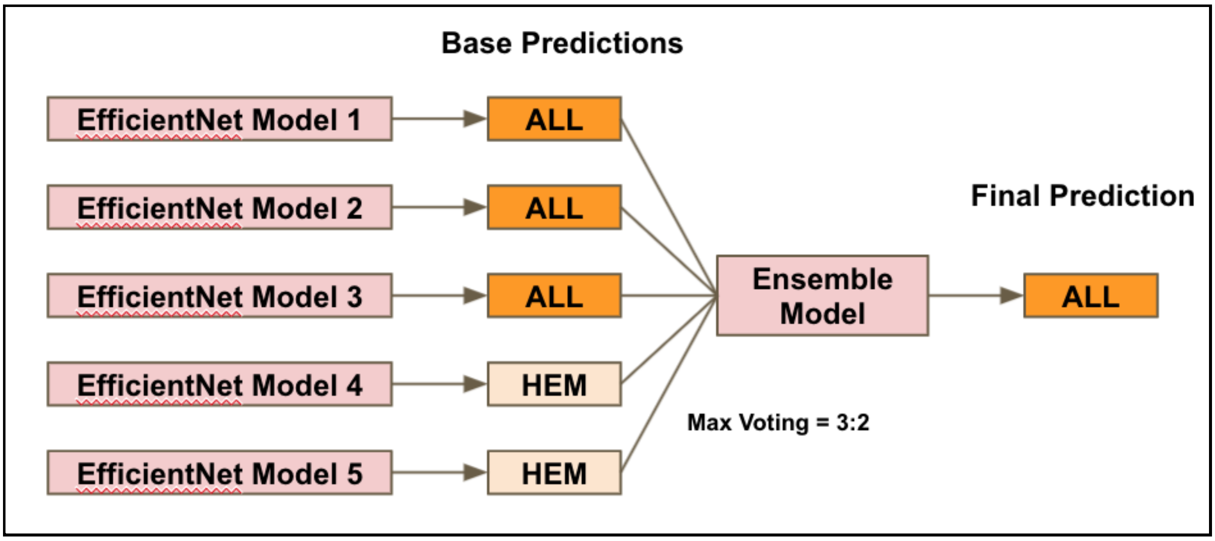


Figure 2 Ensemble EfficientNet model using majority voting to obtain a final prediction for an image. The ensemble model outputs the base prediction that has the most votes as the final prediction.

Evaluating Models

Accuracy, sensitivity, specificity, positive prediction value (PPV) and F1 scores were chosen as performance metrics for model evaluation. The accuracy of a model indicates the total proportion of cells correctly identified, sensitivity measures the percentage of correctly classified ALL cells, specificity measures the percentage of correctly classified healthy blood cells and PPV measures the proportion of cancer cells correctly classified. Accuracy scores are influenced by underlying data distributions, where models can achieve high accuracies even when providing non-useful predictions for over-represented classes. Unlike accuracy values, F1 scores are non-biased towards unevenly-distributed datasets similar to our testing set

(HEM:ALL = 8:17) (Table 1). Thus, the F1 score is used as an additional metric to estimate our model performance.

The performance metrics used for evaluation are defined as follows:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

$$\text{Sensitivity} = \frac{TP}{TP+FP}$$

$$\text{Specificity} = \frac{TN}{TN+FN}$$

$$\text{PPV} = \frac{TP}{TP+FN}$$

$$\text{F1 score} = \frac{2(PPV \times Sensitivity)}{PPV+sensitivity}$$

where TP, TN, FP and FN refers to true positives, true negatives, false positives and false negatives respectively.

Code Availability

The Leukaemia Image Classification project is carried out in Python 3.6.9 using Google Colaboratory. All datasets, scripts and results are made available on Github via https://github.com/Amandahsr/ZB4171_LeukaemiaImageClassification.git.

3. EfficientNet Model

Finetuning hyperparameters of a neural network model is often done in an arbitrary manner to obtain a "best performing" model. However, the number of parameters to finetune increases exponentially as model complexity increases, making the task of finding optimal parameters especially challenging and inefficient. Additionally, arbitrary parameter tuning typically lead to sub-optimal model performance. The authors of EfficientNet, *Tan and Le,* found that balancing the three dimensions of depth, width and resolution can significantly improve performance when integrated with CNN models such as ResNet or Inception (Tan & Le). The EfficientNet model architecture leverages upon the idea of balanced scaling by integrating a new scaling method known as uniform scaling. Uniform scaling is carried out using compound coefficient and integrates a systematic way of scaling up dimensions by carrying out grid search subjected to additional constraints.

Depth refers to the number of layers in the network, where deeper networks tend to capture more complex features and generalises better on new tasks. Width refers to the number of feature maps per layer in the network. The inclusion of more feature maps allows the model to capture more fine-grained features while keeping the architecture small and easy to train. Resolution refers to the height and width of input images. The higher the resolution of input images, the more detailed the extracted features are during training. EfficientNet balances these dimensions by scaling with a constant ratio, where the constant ratio is determined by parameters α, β and γ, representing each of the 3 dimensions. These parameters are exponentiated by a φ term, denoting the amount of computational resources available:

$$\text{Depth: } d = \alpha^{\varphi}$$

$$\text{Width: } w = \beta^{\varphi}$$

$$\text{Resolution: } r = \gamma^{\varphi}$$

To ensure that the model complexity does not exceed a maximal threshold, EfficientNet searches for the optimal parameters by carrying out grid search subjected to the following constraints:

$$\alpha^{\varphi} \times \beta^{\varphi} \times \gamma^{\varphi} \approx 2$$

A baseline model EfficientNet-B0 is obtained by fixing φ = 1 and carrying out a small grid search of α, β and γ on ImageNet. The EfficientNet-B0 model was chosen as a pre-trained model in our project for further scaling during training on our leukaemia data. EfficientNet significantly outperforms other high-accuracy CNNs on ImageNet while having a smaller or equivalent size, making it a suitable model to implement for this project considering the computational constraints faced using Google Colaboratory.

4. Noisy Student Training

Noisy Student training is a semi-supervised method built to further improve model performance. Models trained using Noisy Student achieve top 1% accuracy when tested with ImageNet and have significant gains in robustness (Xie et al., 2020). The Noisy Student incorporates self-training with equal-or-bigger student models and noisy data during the training process. The training method follows a 4-step process for each iteration (Figure 3):

1. Train a teacher model on a set of labelled images.
2. Predict the labels of a set of unlabelled images using the teacher model, generating pseudo-labelled images.
3. Train a student model on the set of pseudo-labelled images.
4. Initialise the student model as the new teacher model and repeat from step 2.

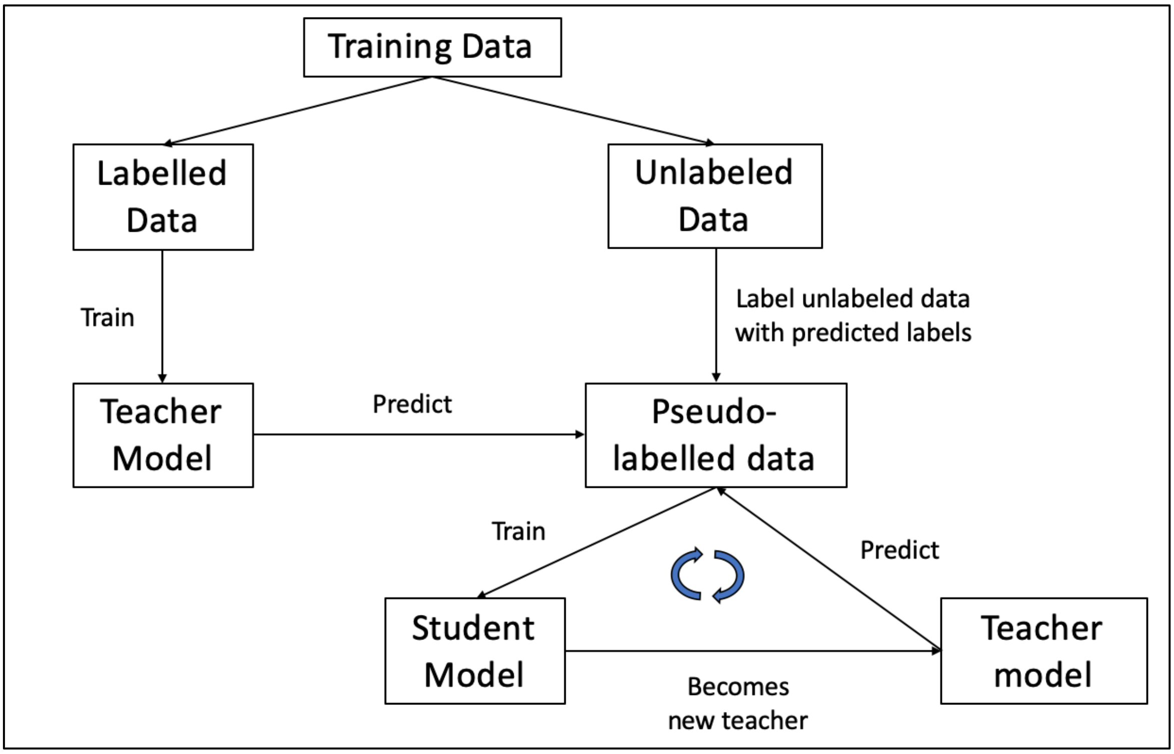


Figure 3 Overview of Noisy Student training. Each Noisy Student iteration involves training a teacher model, generating pseudo-labelled images, training a student model and repeating the process with the student model initialised as the new teacher model.

The student model in the final iteration of Noisy Student training is taken as the final trained Noisy Student model. Noisy Student training can significantly improve the performance of EfficientNet due to significant noise added to the training process. Noise is added via the set of pseudo-labelled images when the teacher model predicts the wrong labels for the unlabelled set. Training on noisy pseudo-labelled images forces the student model to train harder and predict more accurately as the new teacher model in the next iteration. With added noise, the models trained using Noisy Student is less likely to overfit on training data and generalise better when predicting new images. In addition, the training set used for this project contained relatively few images (~10,000) and data augmentation was not possible due to computational constraints. The Noisy Student model generates more unseen data via the pseudo-labelled images and producing better performing models with more images.

5. Results and Discussion

K-Fold Cross Validation with Nested Epoch Training

A training loop of 5-fold cross validation with 50 epochs nested in each fold is carried out to train EfficientNet and Noisy Student models. K-fold cross validation reduces bias and variance because every image in the training set is used once in the validation set, providing a better estimate of model performance. Each fold trains and validates a model on a training and validation split, producing in total, 5 EfficientNet and 5 Noisy Student models.

Validation results for the 5 trained EfficientNet models show an overall decrease in validation loss and increase in validation accuracy, indicating that the models are producing more accurate predictions after each epoch training (Figure 4). The validation curves also plateau towards the end of 50 epochs, suggesting that the trained models have not overfit or underfit on the training splits. In addition, testing loss and accuracies across folds were similar, indicating that the training and validation sets were well split for each fold. Overall, k-fold cross validation with nested epochs produced well-trained EfficientNet models with test accuracies averaging ~86%.

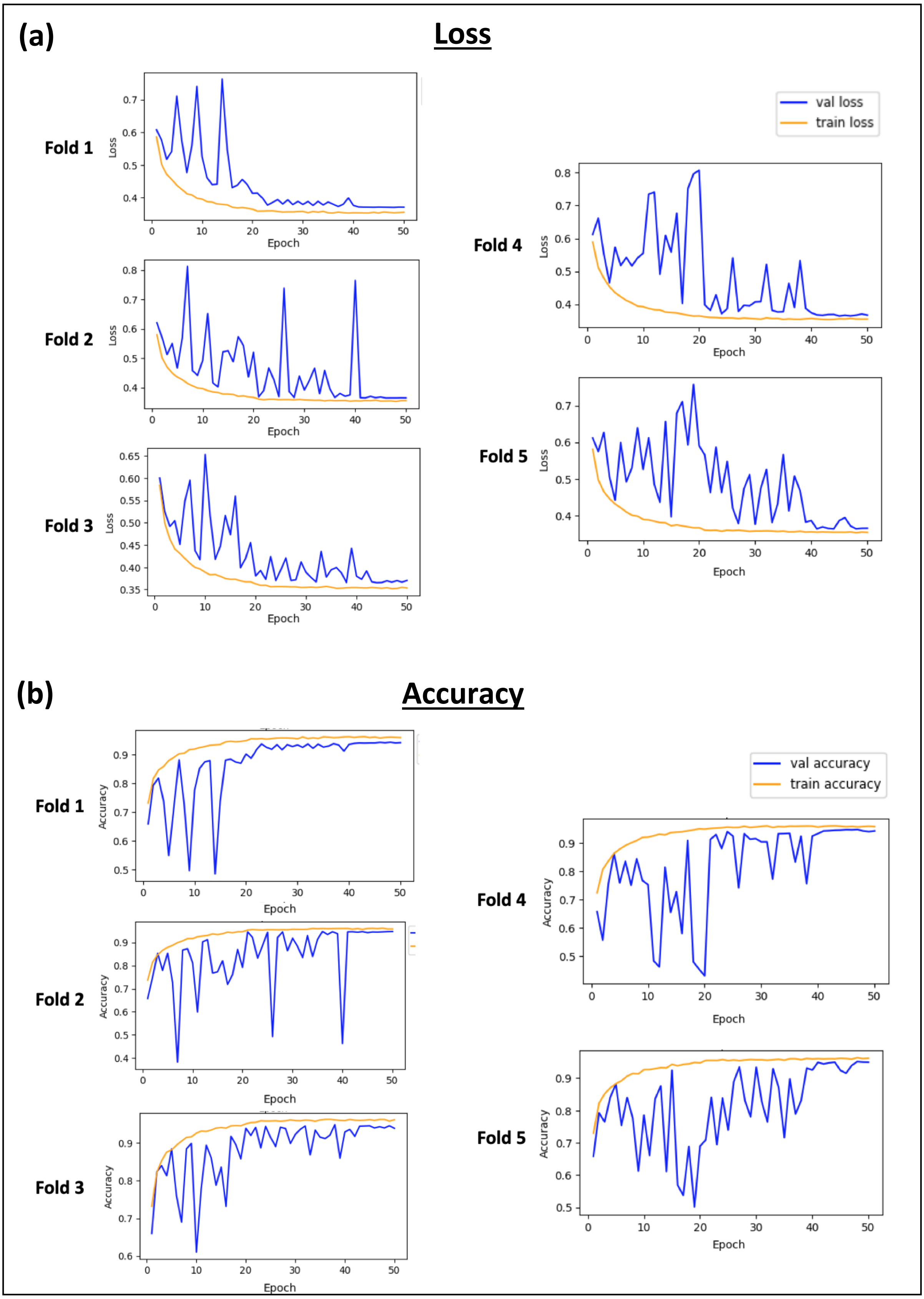
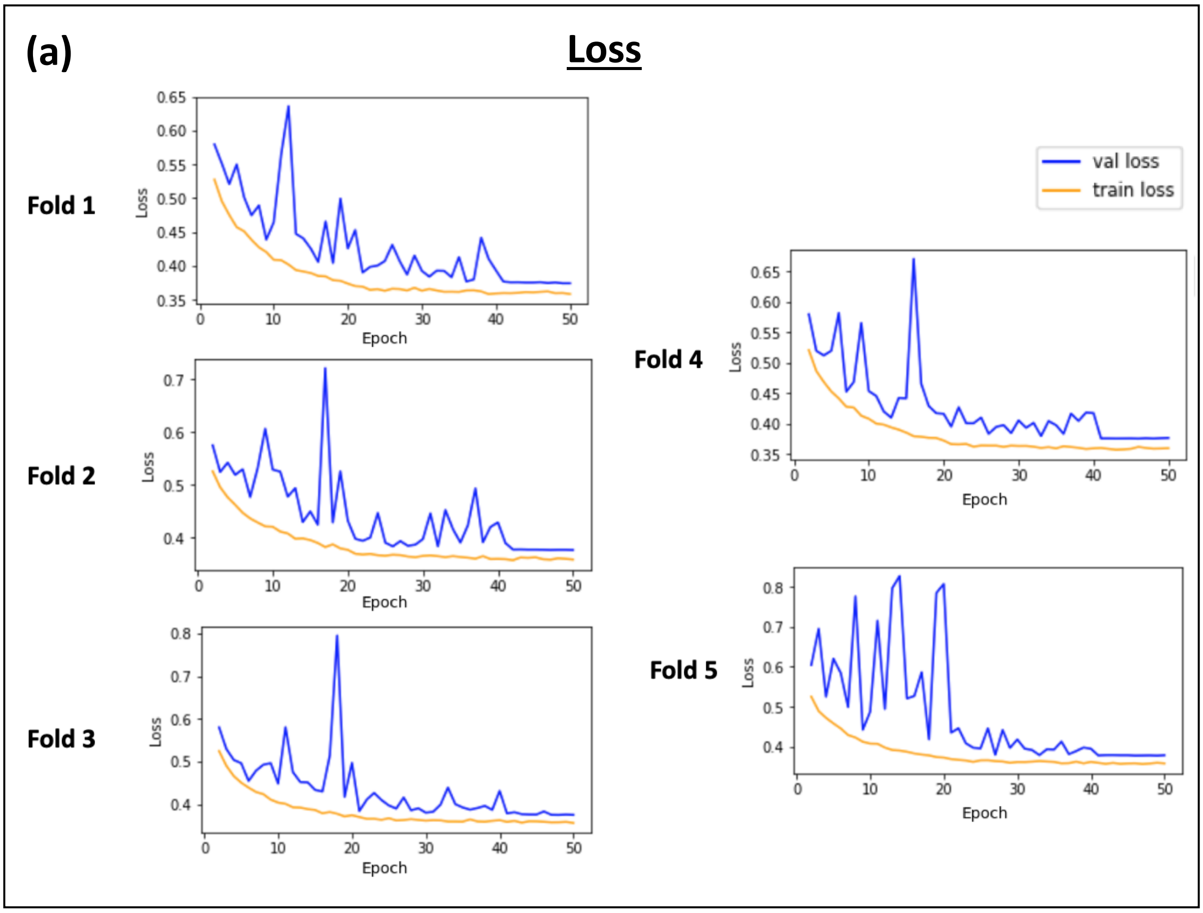


Figure 4 (a) Loss and **(b)** Accuracy of EfficientNet models against epoch number in each fold of k-fold cross validation in training and validation set.

Similar validation results were obtained for the 5 trained Noisy Student models after 50 epochs per fold (Figure 5). The validation loss and validation accuracy showed an overall decrease and increase respectively, indicating more accurate predictions obtained by the Noisy Student models after each epoch training. The validation curves also plateau towards the end of 50 epochs, suggesting well-trained models that have not overfit or underfit on the training splits. In addition, testing loss and accuracies across folds were similar, indicating that the training and validation sets were well split for each fold. Overall, k-fold cross validation with nested epochs produced well-trained Noisy Student models with test accuracies averaging ~86%.



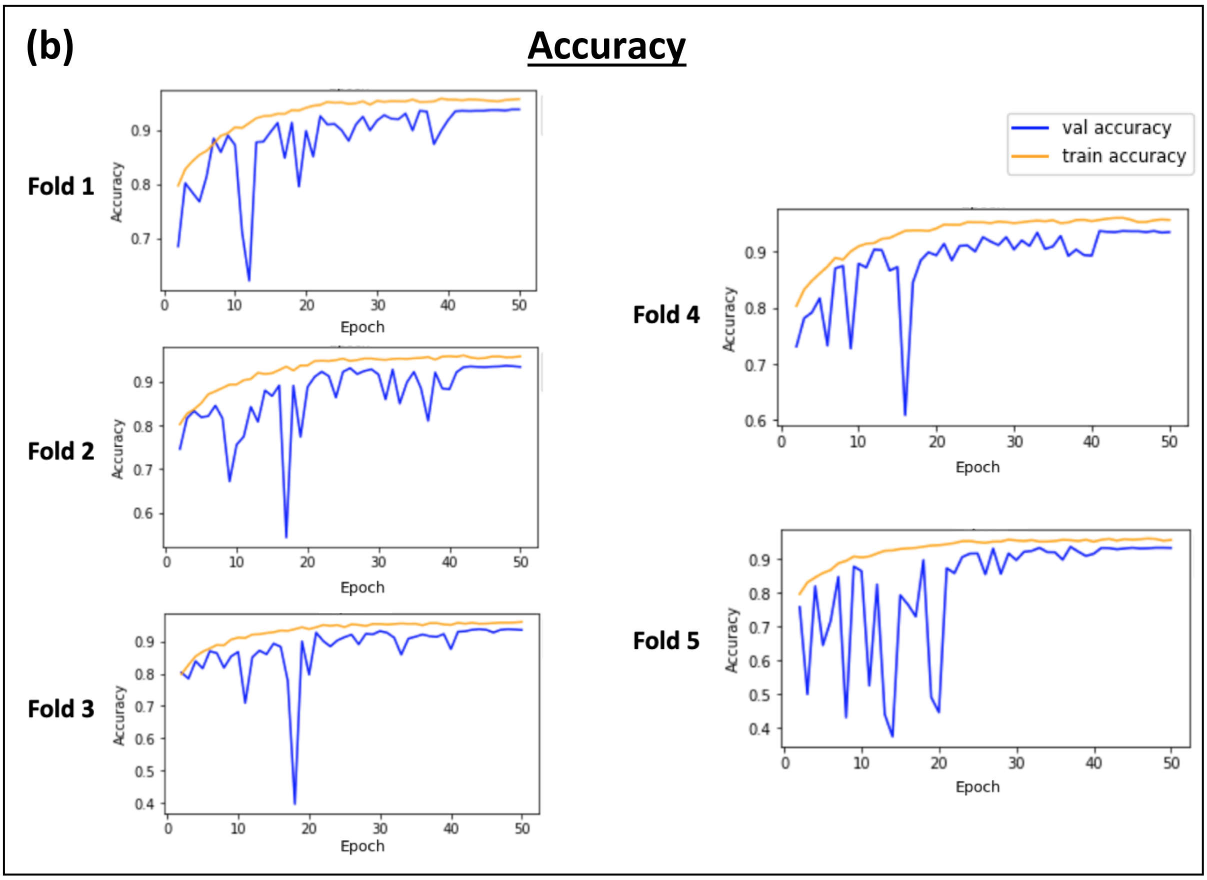


Figure 5 (a) Loss and **(b)** Accuracy of Noisy Student models against epoch number in each fold of k-fold cross validation in training and validation set.

Limited Improvement in Model Performance with Noisy Student Training

Noisy Student training was implemented to further improve EfficientNet model performance. Unexpectedly, the Noisy Student models performed similarly across the performance metric to the EfficientNet models (Table 2). We suspect that the training parameters of the Noisy Student training were sub-optimal, thus producing little to no improvement on model performance. Due to computational constraints, only 3 Noisy Student training iterations were implemented in the training loop. As a result, improvement in model performance may also be limited by the number of Noisy Student iterations implemented. Nevertheless, it is optimistic that the Noisy Student models were able to obtain similar model performance as the EfficientNet models with only 3 Noisy Student iterations. Future work focused on tuning training hyperparameters of the Noisy Student training, such as number of iterations and number of epochs for teacher and student training, should be explored to exploit the full potential of the Noisy Student method.

Table 2 Test Accuracies, sensitivity, specificity, PPVs and F1 scores of EfficientNet (EN) and Noisy Student (NS) models.

	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Mean**	**Variance**
EN Accuracy (%)	86.61	86.54	85.87	86.50	86.21	86.34	0.09
NS Accuracy (%)	86.87	86.41	86.99	86.00	85.09	86.30	0.59
EN Sensitivity (%)	89.49	89.65	88.94	89.16	89.16	89.28	0.08
NS Sensitivity (%)	89.43	89.49	89.81	89.11	87.8	89.16	0.61
EN Specificity (%)	77.51	76.64	76.12	78.03	76.81	77.02	0.56
NS Specificity (%)	78.72	76.64	78.03	76.12	76.47	77.20	1.25
EN PPV (%)	92.67	92.42	92.21	92.80	92.43	92.51	0.05
NS PPV (%)	93.03	92.41	92.85	92.22	92.22	92.55	0.14
EN F1 score (%)	91.05	91.01	90.55	90.94	90.76	90.86	0.04
NS F1 Score (%)	91.2	90.92	91.31	90.63	89.96	90.82	0.29

Ensemble Framework Improves Model Performance

An ensemble model typically produces more accurate predictions and are more robust than individual models. To further improve model performance, an ensemble framework was implemented using the 5 EfficientNet and 5 Noisy Student models after training is completed.

The Ensemble EfficientNet and Ensemble Noisy Student models use majority voting as a voting system to output a final prediction from base predictions. We found that the Ensemble EfficientNet model significantly outperformed its base models across all performance metrics (Table 3). The Ensemble Noisy Student model outperformed its base models in accuracy, sensitivity and F1 scores while obtaining similar PPVs and a reduced specificity (Table 3). Overall, the ensemble framework improved model accuracies by ~3% for both EfficientNet and Noisy Student models.

Table 3 Mean test accuracies, sensitivity, specificity, PPVs and F1 scores of EfficientNet, Noisy Student, Ensemble EfficientNet and Ensemble Noisy Student models.

Model	**Mean Accuracy (%)**	**Mean Sensitivity (%)**	**Mean Specificity (%)**	**Mean PPV (%)**	**Mean F1 Score (%)**
EfficientNet Models	86.34	89.28	77.02	92.51	90.86
Noisy Student Models	86.30	89.16	77.20	92.55	90.82
Ensemble EfficientNet Model	89.11	90.14	85.81	95.28	92.64
Ensemble Noisy Student Model	90.93	95.86	75.26	92.49	94.14

Interestingly, the ensemble framework benefitted the sensitivity of the Noisy Student models greater than the EfficientNet models, while improving the specificity of the EfficientNet models greater than the Noisy Student models. The Ensemble Noisy Student model had a ~6% increase in sensitivity compared to the Noisy Student models while the Ensemble EfficientNet model had a ~8% increase in specificity. This suggests that the ability of the ensemble framework, or the majority voting system used in our ensemble models, to improve base model performance may be dependent on the base model architecture used.

Ensemble Noisy Student Model best suited for ALL detection

It is ideal that diagnostic tools are well-rounded and perform well across all performance metrics. However, it is challenging to design a tool that is able to detect both true positives and true negatives accurately. In the context of ALL detection, the ability to detect true positives is more important than the ability to detect true negatives because undetected ALL cases are more severe than undetected healthy cases. Thus, image classification models with higher sensitivities for ALL detection could be more useful than those with higher specificities. The Ensemble Noisy Student model is thus best suited for the task of ALL detection because it obtained the best performance in accuracy (~90%), F1 score (~94%) and sensitivity (~95%) among all the models implemented. The model also obtained a relatively high specificity (~75%) and PPV (~92%), indicating that it is well-rounded and suitable to use for ALL detection.

Project Limitations

The training and validation sets obtained from the Kaggle challenge were relatively small, containing a total of ~10,000 images after merging the two sets. Due to memory constraints on Google Colaboratory, data augmentation was not carried out to generate additional images for training and testing. With limited number of images available for training, the number of epochs per k-fold cross validation and the number of Noisy Student training iterations carried out were kept small. Further work is required on a platform with bigger memory capacity for data augmentation, increased epochs per fold and increased Noisy Student training iterations to be carried out to produce better performing models.

The Noisy Student model was implemented to further improve EfficientNet model performance. However, student models will not surpass teacher models if pseudo-labels generated by teacher models are inaccurate during student training. This phenomenon is also known as the problem of confirmation bias in pseudo-labeling (Arazo et al., 2020). Another CNN model known as Meta Pseudo Labels (MPL) also capitalizes on pseudo-labelling to train a student model, but implements reinforcement learning by adding an additional feedback loop to reward the teacher model for accurate pseudo-labelling (Pham et al., 2021). MPL was initially implemented for EfficientNet improvement, but was not successfully carried out due to computational and memory constraints on Google Colaboratory. Further work is required to test other training methods such as MPL to produce more accurate EfficientNet models.

6. Conclusion

We have successfully implemented an image-based classification model using the EfficientNet architecture, with further gains in model accuracy using an ensemble framework and Noisy Student training. Our Ensemble Noisy Student model obtained a high F1 score of 94.14%, outperforming the current best performing model (~83%) submitted to the Leukemia Classification challenge on Kaggle. For further improvements, data augmentation techniques and different methods of self-training with pseudo-labels can be implemented together with our Ensemble Noisy Student model. Existing work can also be extended to building a multi-classification model for diagnosing ALL subtypes or other leukaemia types such as Acute Myeloid Leukaemia (AML). Our project reveals the potential of AI tools in clinical diagnostics and the ability of ML techniques to facilitate ALL diagnosis critical for patient survival. With a robust and accurate AI system, ML models similar to ours can resolve issues in cancer diagnostics and make cancer diagnosis more efficient and precise.

7. References

Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., & McGuinness, K. (2020). *Pseudo-Labeling and Confirmation Bias in Deep Semi-Supervised Learning*. Paper presented at the 2020 International Joint Conference on Neural Networks (IJCNN). arXiv:1908.02983v5

Chandradevan, R., Aljudi, A. A., Drumheller, B. R., Kunananthaseelan, N., Amgad, M., Gutman, D. A., . . . Jaye, D. L. (2020). Machine-based detection and classification for bone marrow aspirate differential counts: initial development focusing on nonneoplastic cells. *Lab Invest, 100*(1), 98-109. doi:10.1038/s41374-019-0325-7

Gupta, A., Duggal, R., Gehlot, S., Gupta, R., Mangal, A., Kumar, L., . . . Satpathy, D. (2020). GCTI-SN: Geometry-inspired chemical and tissue invariant stain normalization of microscopic medical images. *Med Image Anal, 65*, 101788. doi:10.1016/j.media.2020.101788

O, R., J, D., H, S., J, K., S, S., S, M., . . . AC, B. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision, 115*(3), 211-252.

Pham, H., Dai, Z., Xie, Q., & Le, Q. (2021, 1 March 2021). *Meta Pseudo Labels.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Tan, M., & Le, Q. (24 May 2019). *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks.* Paper presented at the International Conference on Machine Learning.

Terwilliger, T., & Abdul-Hay, M. (2017). Acute lymphoblastic leukemia: a comprehensive review and 2017 update. *Blood Cancer J, 7*(6), e577. doi:10.1038/bcj.2017.53

van der Laak, J., Litjens, G., & Ciompi, F. (2021). Deep learning in histopathology: the path to the clinic. *Nat Med, 27*(5), 775-784. doi:10.1038/s41591-021-01343-4

Xie, Q., T., L. M., Hovy, E., & V., L. Q. (2020, 19 June 2020). *Self-training with Noisy Student improves ImageNet classification.* Paper presented at the Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.

Yu, K. H., Beam, A. L., & Kohane, I. S. (2018). Artificial intelligence in healthcare. *Nat Biomed Eng, 2*(10), 719-731. doi:10.1038/s41551-018-0305-z