ABSTRACT

This paper is to explore “transfer learning” as a way to effectively improve convolutional neural network (CNN) architectures in computer vision applications. Alternative network structures/topologies between fine-tuned convolutional neural networks with subnets are explored. The key assignment objective is to further assess and refine CNNs through transfer learning as an extension to *Assignment 2*. Management recommendation is provided for making use of pretrained models (i.e., transfer learning) to springboard deep learning projects.

1. INTRODUCTION

In this exercise, deep neural network topologies combining CNN “subnet” trained on ImageNet data and additional hidden layer(s) are built and tested in the Python coding framework. Attempts were made to compare and evaluate alternative “subnets” structures including MobileNet, DenseNet, and DenseNet with additional dense layers. All models operate in two phases: first the transferred CNN is “frozen” to only train the additional dense layer(s), then the transferred layers are unfrozen to allow fine-tuning of the entire model, which is a different mode of CNN operation than was explored in assignment 2 previously. As CNN are particularly adept in computer vision tasks, the research topic of this paper is to again conduct image classification across all experiments with the aid of transfer learning, the action of taking features learned on one problem, and leveraging them on a new, similar problem which is image classification in this paper.

The CNN architecture variations are examined in this paper for each of their training requirement and performance in classifying the HAM10000 dataset (Tschandl, Rosendahl and

Kittler, 2018), a large collection of multi-source dermatoscopic images of common pigmented skin lesions presented in the ISIC 2018 challenge. As detailed in the metadata table, the dermatoscopic images were sourced from different populations, acquired and stored by different modalities. The final dataset consists of 10015 images across 7 classes, a representative collection of all important diagnostic categories in the realm of pigmented lesions: Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec), basal cell carcinoma (bcc), benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses, bkl), dermatofibroma (df), melanoma (mel), melanocytic nevi (nv) and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, vasc) (Tschandl, 2018). To note, the dataset includes lesions with multiple images, which can be tracked by the "lesioned"-column within the HAM10000_metadata CSV file. The 7 classes are imbalanced, and appropriate preprocessing treatment of inputs is discussed in the Methods section. Extensive exploratory data analysis (EDA) was conducted on the dataset before running the experiments, which will be further detailed in the Methods and Results sections of this paper.

After the EDA was completed, 4 modeling experiments were conducted: Experiment 1 and 2 trained different versions of MobileNet architecture with an additional dense layer and regularization, whereas Experiment 3 and 4 are the same DenseNet201 architecture paired with different number of dense layers and regularization. At a minimum, all models had the training/validation metrics plotted and classification predictions on the test set summarized via confusion matrices.

In addition, the model in Experiment 3 with the "best performance" among those tested had the input feature subject to post-training analysis by a technique called LIME (Local

Interpretable Model-Agnostic Explanations), discussed briefly in the Literature Review and Results sections.

This assignment is designed to practice advanced techniques of building CNN by taking advantage of "subnets" pretrained on large-scale image inputs (i.e., ImageNet), therefore acting as excellent image feature extractors to generalize on the specific image dataset in scope. More importantly, the principles of transfer learning are leveraged to provide management recommendation in machine learning project strategies.

2. LITERATURE REVIEW

Introduction of CNN structure and operation by Walkarn (2016), Challot (2021), Deshpande (2016) were reviewed in Assignment 2. MobileNet operational theories are presented by Howard et al. (2017). MobileNetV2 and MobileNetV3 are presented by Sandler et al. (2018) and Howard et al. (2019) respectively. MobileNetV2 is based on "an inverted residual structure where the residual connections are between the bottleneck layers. The intermediate expansion layer uses lightweight depthwise convolutions to filter features as a source of non-linearity". MobileNetV3 is built on MobileNetV2 with squeeze and excite in the residual layer, where -Large and -Small are targeted for high and low resource use cases. DenseNet architecture was proposed by Huang et al. (2017). LIME algorithm was developed by Ribeiro et al. (2016).

3. METHODS

In the EDA stage, Python packages including NumPy, Pandas, and Matplotlib are used to help with data manipulation and visualization. Input image metadata are examined with the Pandas "read_csv" function. The input features are displayed as columns in the loaded

dataframe. Three additional columns - “Path” to store the image file paths, “Cell_type” to explain “dx” in interpretable class labels, and “Cell_type_idx” to convert “Cell_type” into integer labels for model training later - are created as engineered features. Five scaled down images of dimension 125x100 in each of the 7 skin lesion categories (figure 1) are viewed as samples. The numerical features are summarized in a preliminary statistical analysis. Data of “Age” feature was cleaned up by filling the null values with its mean value in the dataset. Both univariate and bivariate analysis are performed on the input features and label class data by selectively plotting and examining bar charts, pie charts, and count plots, which are discussed in details in the Results section. Observed class imbalance (figure 2) is also separately addressed in the Results section.

The full HAM10000 dataset is both available on Harvard Dataverse (Tschandl, 2018) and Kaggle (*Skin Cancer MNIST: HAM10000*, 2018). Using OpenCV packages in a custom function, all 10015 skin lesion images are loaded and subsequently resized to (128,128,3) weight x height x channel, with the RGB pixel values normalized (i.e., divided by 255 to convert values to a range between (0,1)). Seen from figure 2, over 65% of images are labelled in one lesion class: Melanocytic nevi. Before the CNN experiments were run, the training subset, validation subset and test subset were split from the full dataset in a 60:20:20 ratio.

The CNN models were built, trained, and tuned in a custom “Tuner” function that wraps the pretrained CNNs and additional dense layers utilizing a variety of Keras packages imported from Tensorflow. Sourcing from Keras.applications module, the premade CNN architectures with pre-trained weights, including MobiletNetV2, MobileNetV3Large/Small, and DenseNet201 are used as the “transferred” base layers. The additional dense layers are regularized by Dropout and

Batch normalization. As for data preprocessing, “data oversampling”, “data augmentation”, and “class weights setting” are comprehended in custom functions as attempts to tackle the imbalanced class data and the lack of certain class labels. To facilitate a fair comparison of model performance, some parameters were held constant across different models as listed in Table 1. In Experiment 1 structure, the pretrained MobileNetV2 output (4,4,1280) channels, flattened and fed into a fully-connected layer of 128 nodes after dropout. In Experiment 2 structure, the pretrained MobileNetV3Large output (4,4,960) channels, flattened and fed into a fully-connected layer of 128 nodes after dropout. In Experiment 3 and 4 structure, the pretrained DenseNet201architecture output (4,4,1920) channels, flattened and fed into a fully-connected layer of 128 nodes (Experiment 3) and two fully-connected layers of [512 -> 128] nodes (Experiment 4). All models are completed with a dense classification layer with 7 nodes that correspond to the seven image classes with a Softmax activation.

Model fine-tuning is made possible because the Tuner function incorporates a method to compare the loss function during each epoch in both phases. The best model weights are saved as checkpoint to be applied to the final model after training is completed. Moreover, a class weight setting function is used to pass heavier weights through a parameter to classes that have fewer labels during model input.

The model training and validation performance of each model is inspected by plotting the evolution of accuracy and loss in figure 3 as each epoch completes. To further the analysis, confusion matrices are constructed together with classification metrics report for selected experiments to visualize the occurrence each predicted value against the true value of the label.

4. RESULTS

From the univariate analysis (figure 4) conducted during EDA prior to the experiments, skin lesions are found to be most frequent in the studied population around age 45, and least frequent for age 10 and below. It is also observe that the probability of having skin lesions increases with age. It is slightly more prominent in men as compared to women to have skin lesions, and they appear most commonly on "back" of the body and least common on the "acral surfaces"(such as limbs, fingers, or ears). As stated in the Methods section, the most common skin lesion type is Melanocytic nevi while the least common is Dermatofibroma, contributing to a significant imbalance in the dataset. From figure 5, 90% of the skin lesion instances were identified through histopathology and follow-up examinations.

Digging into the data through bivariate analysis of localization against gender (figure 6) or cell type (figure 7) confirms that the back is the most affected body part among the population and more prominent in men. However, skin lesions on lower extremity of the body are more visible in women. Another interesting observation is that while benign keratosis-like lesions appear the most on the face, other body parts are affected the most by Melanocytic nevi.

During post-training analysis, it was discovered that the original experiments suffer from data leakage as the test subset data was erroneously included in the training subset data. Due to time and resource constraints, only Experiment 1 (MobileNetV2) and Experiment 3 (DenseNet201) can be retrained with proper data hold-out. Therefore, a meaningful comparison was summarized between these two models in Table 2 to compare their model attributes/hyperparameters, model performance, and model processing time/requirement. The model performance metrics plots of these two models are reconstructed in figure 8. Conversely, the original confusion matrices and classification reports are rendered invalid.

From Table 2, there are several key observations. Firstly, both CNNs pretrained on ImageNet generalize quite well over 70% accuracy on the new image data without any fine-tuning. Secondly, the model based on DenseNet201 performs about 4% more accurate than the model based on MobileNetV2 despite a total of 4-fold more model parameters. Thirdly, for both models the fine-tuning phase did not improve the model performance from when the pretrained base model was “frozen”. It is postulated that more dense layers with a larger number of nodes will be required to effectively improve the model performance.

To deal with the class imbalance, aside from assigning class weights, another approach would be to resample the dataset by oversampling the minority class, which is what the custom function “balanced_dataset” is set up to do. Alternatively, data augmentation can also serve a secondary purpose to balance the dataset by filling up minority class with modified images from the same class.

Lastly, useful insights were gained from the interesting model-agnostic technique of LIME. The technique is designed to explain how the input features of any machine learning classifier affect its predictions. It works by perturbing the input and see how the predictions change, which benefit human interpretability because inputs can be perturbed by changing components that make sense to humans. For image classification in this assignment, LIME is applied to divide a correctly predicted skin lesion image into 10 interpretable components (contiguous superpixels) (figure 9). A data set of perturbed instances are then generated by masking some of the superpixels (figure 10). For each perturbed instance, the importance of the perturbance to the original image is computed to learn a locally-weighed linear regression model, eventually

highlight the 4 top superpixels with highest positive weights around the center lesion area and along the right edge as explanations to the class prediction (figure 11).

5. CONCLUSION

This assignment takes deep learning approach to a higher level by emulating transfer learning principles. Transfer learning is especially appropriate on datasets like HAM10000 images where the data size is not significant enough, and allows us to leverage features learned from state-of-the-art models pretrained on large-scale datasets such as ImageNet. No matter the type of model architecture, when transfer learning is adopted in the prediction tasks, it is important to remember to assess and characterize hidden node performance on feature learning for each class detection post-training as a key step to unlock insights and generate interpretable results for us to understand the capabilities and limitations of the model at hand.

As a recommendation to management with respect to deep learning project strategies, a starting point of transfer learning in the context of deep learning is to follow the workflow of the most common incarnation of transfer learning:

1. Take layers and weights from a previously trained model.
2. Freeze them, so as to avoid destroying any of the information they contain during future training rounds.
3. Add some new, trainable layers on top of the frozen layers. They will learn to turn the old features into predictions on a new dataset.
4. Train the new layers on the new dataset.

It is also beneficial to consider including a fine-tuning step which entails unfreezing the entire or part of the model obtained and re-training it on the new data with a *very low* learning rate. In doing so one can potentially achieve meaningful improvements by incrementally adapting the pretrained features to the new data.

Appendix

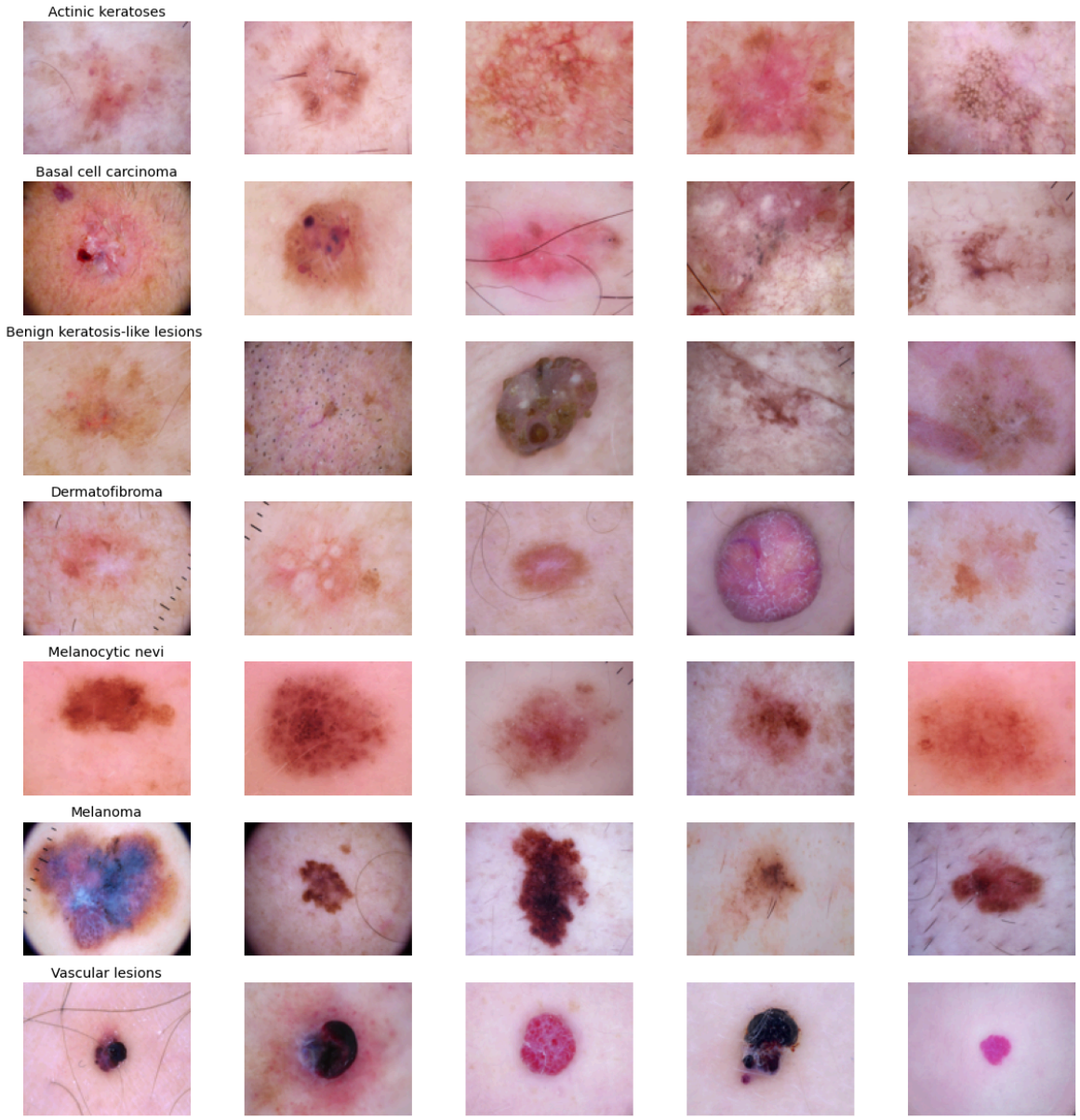


Figure 1. Input image samples for 7 skin lesion class

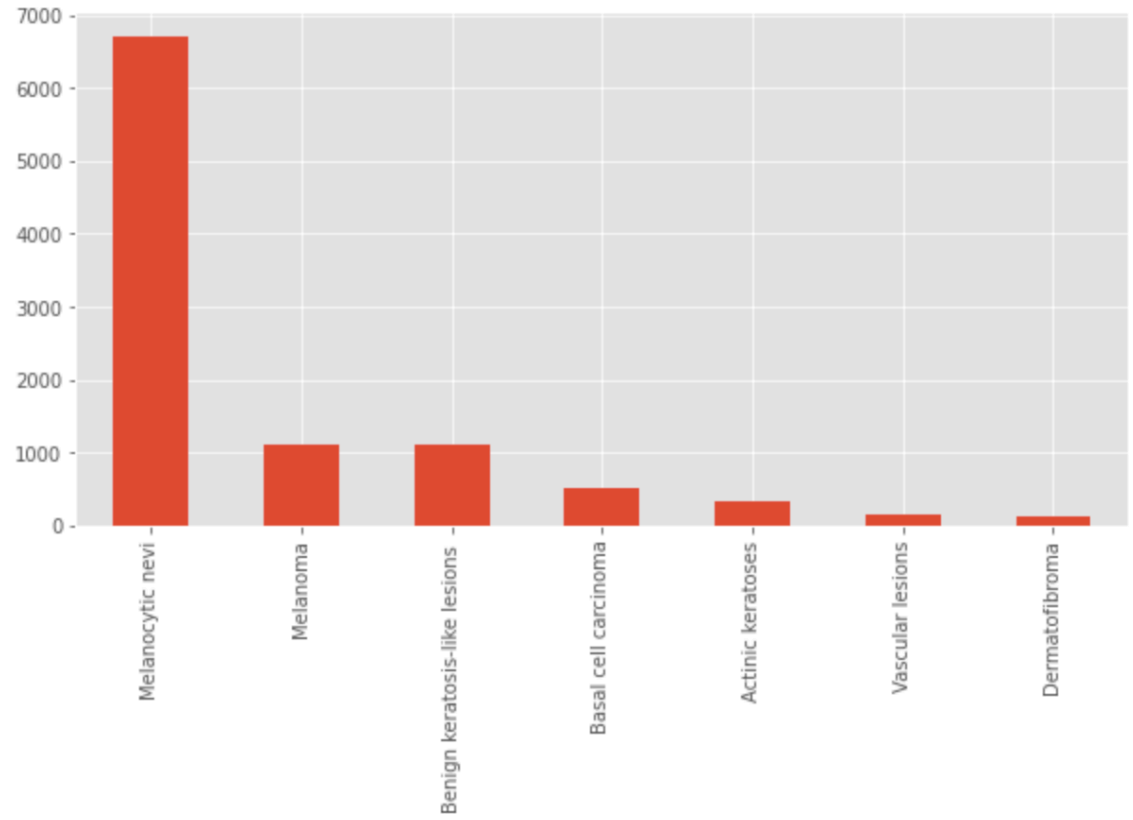
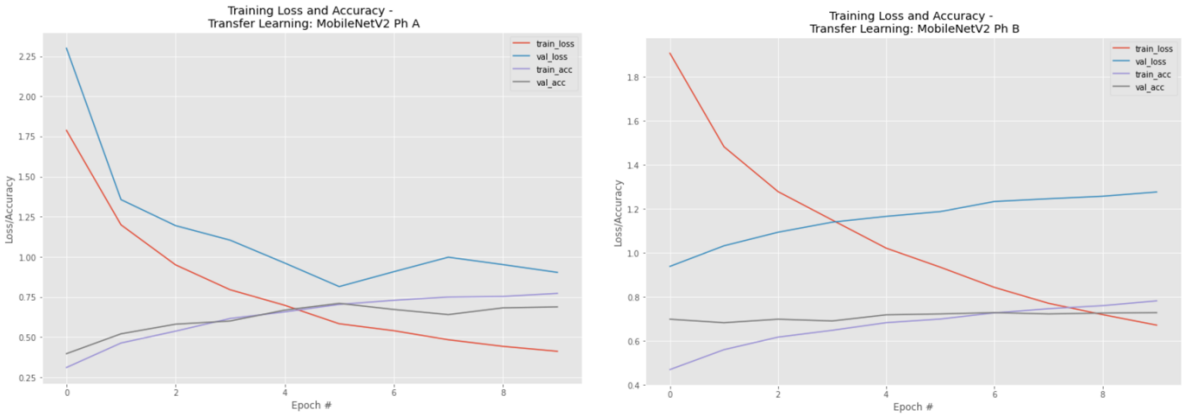


Figure 2. Imbalanced image class distribution



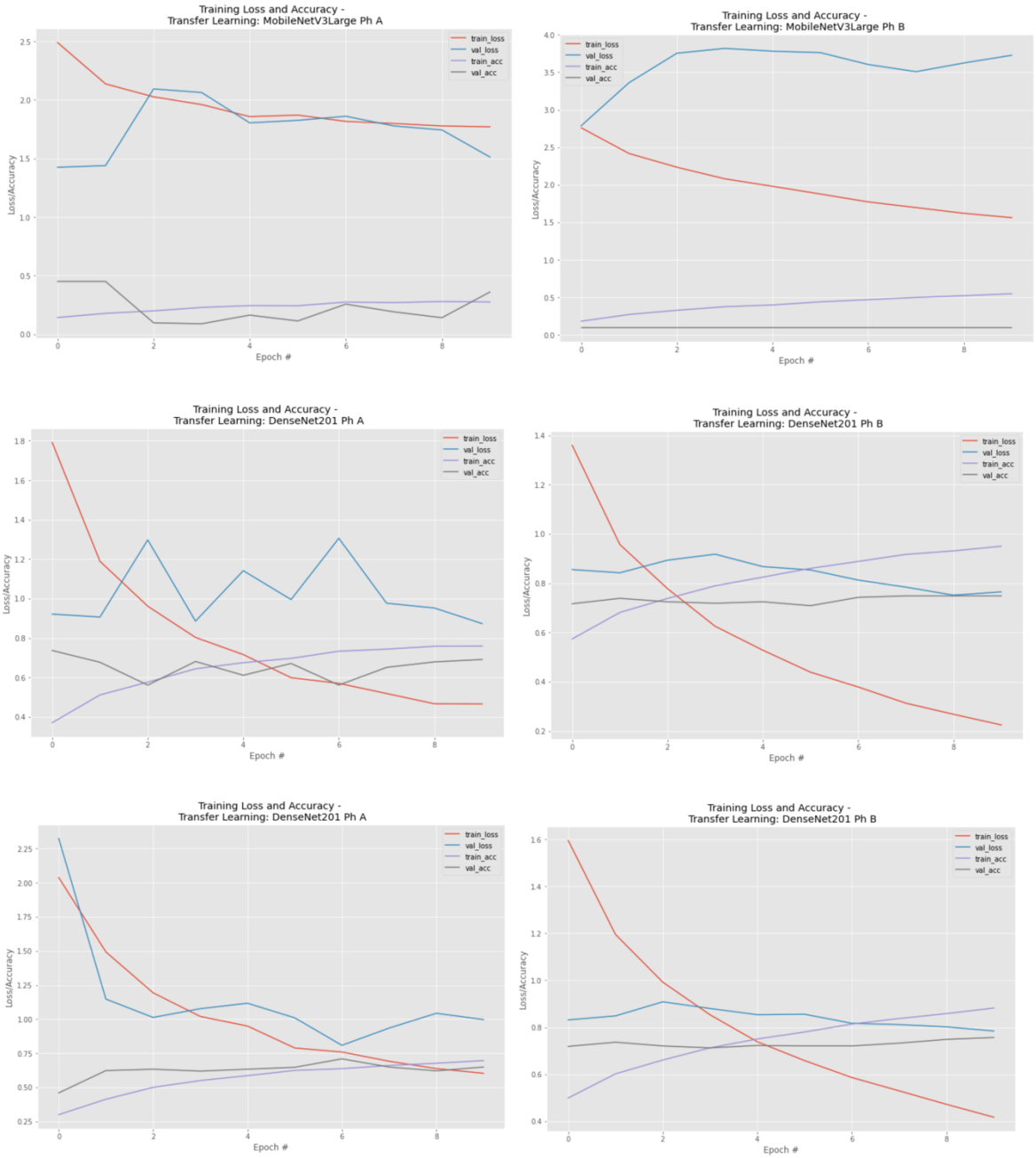


Figure 3. (Top row) MobileNetV2 model performance metrics evolution phase 1 and 2. (2nd row) MobileNetV3Large model performance metrics evolution phase 1 and 2. (3rd row) MobileNetV3Small model performance metrics evolution phase 1 and 2. (Bottom row) DenseNet201 model performance metrics evolution phase 1 and 2.

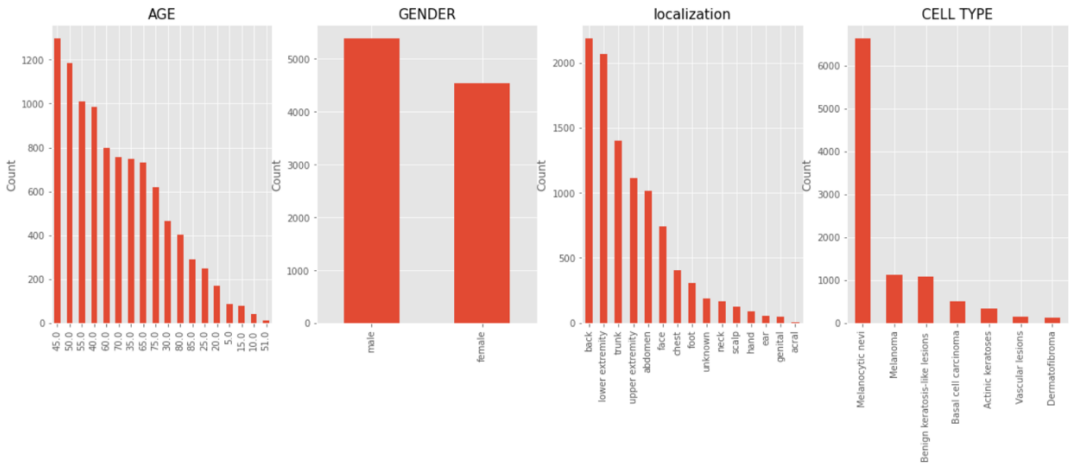


Figure 4. Univariate analysis in EDA

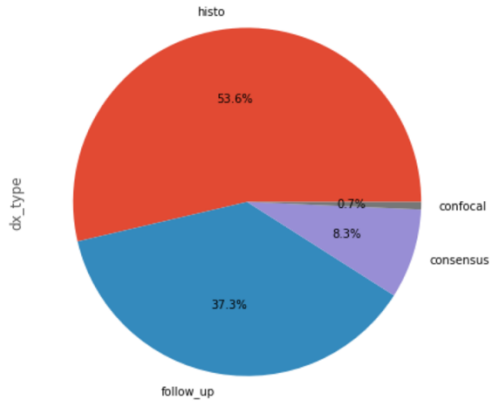


Figure 5. Pie chart of diagnosis method

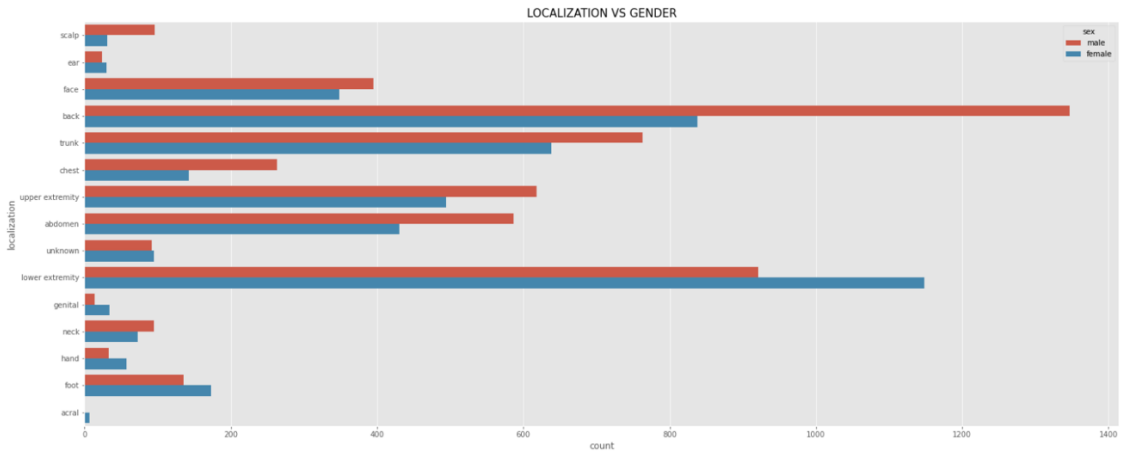


Figure 6. Count plot of localization vs. gender

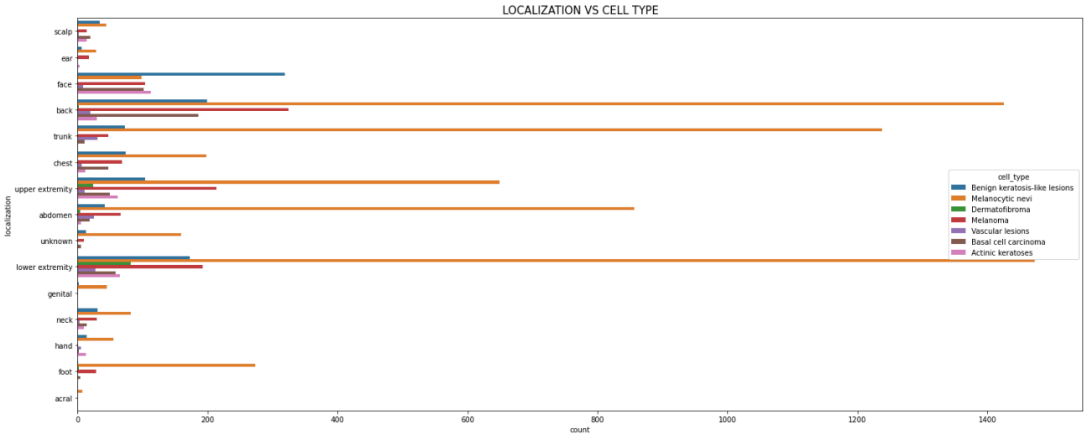


Figure 7. Count plot of localization vs. cell type

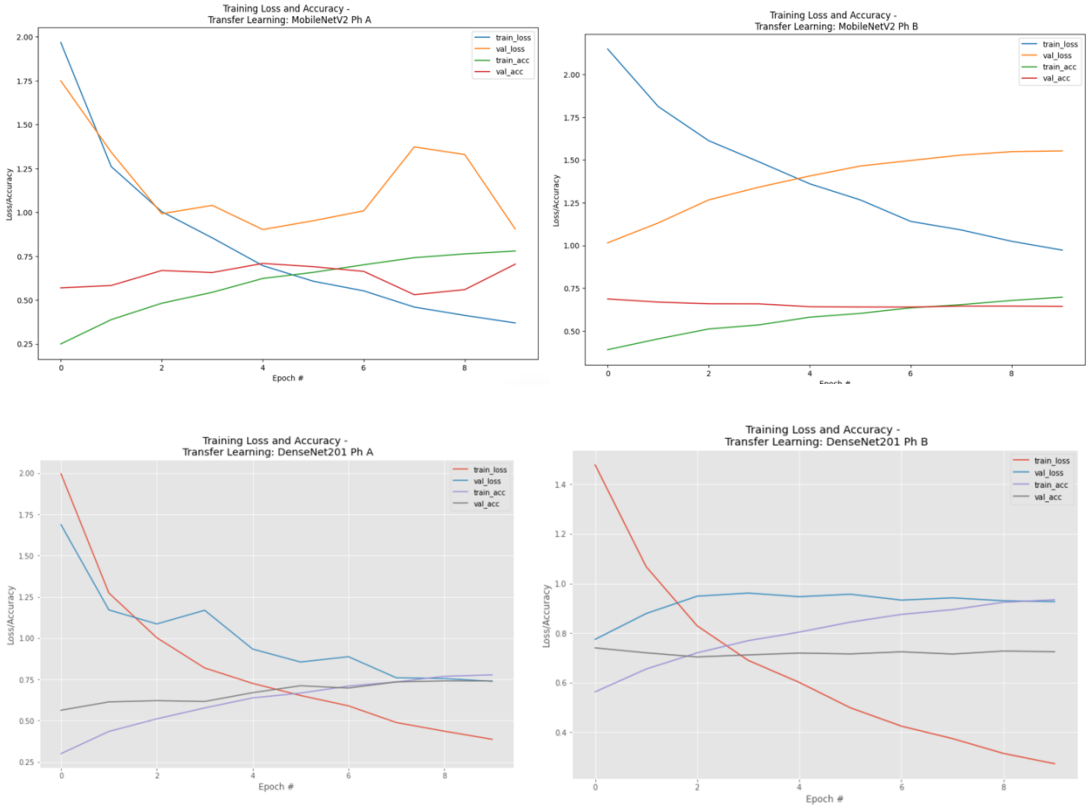


Figure 8. Model performance metrics evolution of MobileNetV2 in phase 1 and 2 (top). Model performance metrics evolution of DenseNet201 in phase 1 and 2 (bottom).

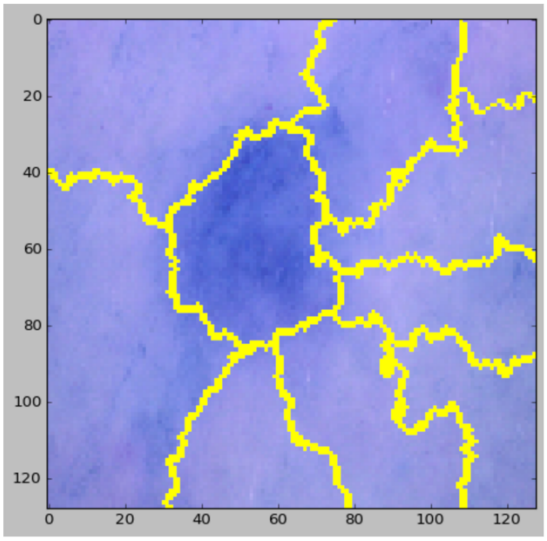


Figure 9. LIME: Superpixel segmentation on label class 4 image

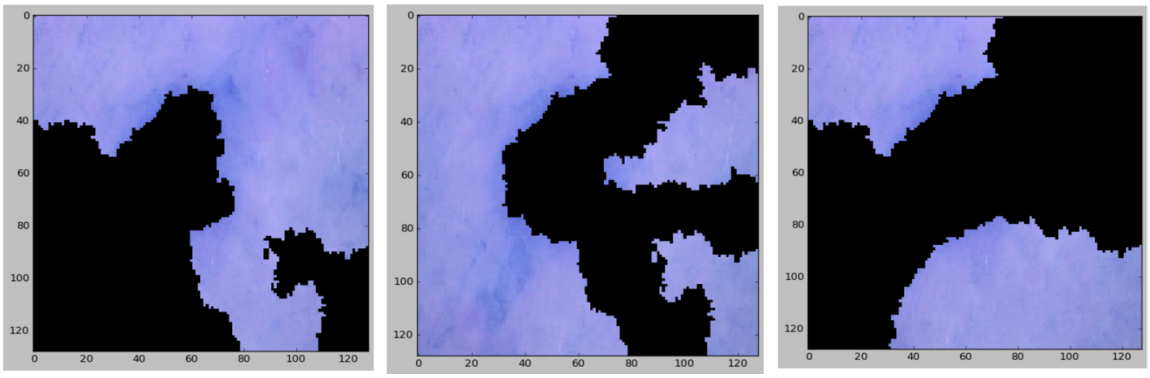


Figure 10. LIME: Superpixel perturbations

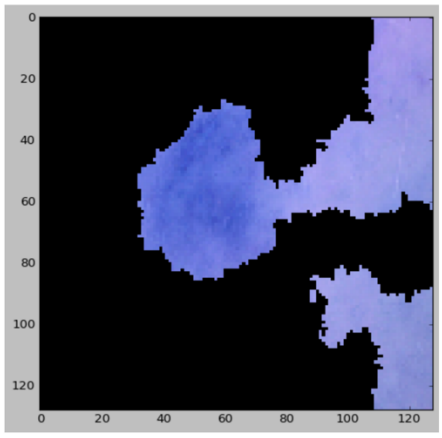


Figure 11. LIME: Top 4 superpixels as input feature explanation to prediction

Epochs	Batch size	Pre-trained weights
10	128	ImageNet

Table 1. Common model parameters across experiments

	Pre-trained architecture	Additional hidden layer and nodes	Regularization	Epochs	Trainable parameters	Total parameters	Training accuracy	Training loss	Validation accuracy	Validation loss	Test accuracy	Test loss
Exp 1	MobileNetV2	1 layer, size 128	Dropout, Batch normalization	10 per phase	2,622,727	4,880,967	62.3%	0.696	70.8%	0.902	70.4%	0.900
Exp 3	DenseNet201	1 layer, size 128	Dropout, Batch normalization	10 per phase	3,933,447	22,255,687	77.8%	0.387	74.1%	0.739	74.1%	0.751

Table 2. Comparison of corrected models in Experiment 1 and 3

References

Tschandl, P., Rosendahl, C., & Kittler, H. (2018). The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific Data, 5(1). https://doi.org/10.1038/sdata.2018.161

Tschandl, P. (2018). *The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions* (Version V3) [Dataset]. Harvard Dataverse. https://doi.org/10.7910/DVN/DBW86T

An Intuitive Explanation of Convolutional Neural Networks. (2016, May 29). Ujjwal Karn. Retrieved October 23, 2022, from https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Chollet, F. (2021, December 21). Deep Learning with Python, Second Edition (2nd ed.). Manning.

Deshpande, A. (n.d.). A Beginner's Guide To Understanding Convolutional Neural Networks. Retrieved October 23, 2022, from https://adeshpande3.github.io/A-Beginner's-Guide-To-Understanding-Convolutional-Neural-Networks/

Andrew Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, M. Andreetto, & Hartwig Adam. (2017b). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. ArXiv: Computer Vision and Pattern Recognition.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). MobileNetV2: Inverted Residuals and Linear Bottlenecks. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. https://doi.org/10.1109/cvpr.2018.00474

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L. C., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., & Le, Q. (2019). Searching for MobileNetV3. 2019 IEEE/CVF International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2019.00140

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://doi.org/10.1109/cvpr.2017.243

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations. https://doi.org/10.18653/v1/n16-3020

Skin Cancer MNIST: HAM10000. (2018, September 20). Kaggle. https://www.kaggle.com/datasets/kmader/skin-cancer-mnist-

ham10000/code?datasetId=54339