

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my teacher, Dr.Nguyen Viet Dung (dung.nguyenviet1@hust.edu.vn) who gave me the golden support to do this wonderful project. He gave a chance to use High GPU computing computer for AI Training. Otherwise, he also gave me recommendation on how to implement experiment to make a good conclusion such as what I should focus on, what I need to investigate, which metrics I should consider.

ABSTRACT

Today, the rapid development of industrial zones leads to an increasing incidence of skin diseases because of polluted air. According to a report by the American Cancer Society, it is estimated that in 2022 there will be about 100,000 people suffered from skin cancer and more than 7600 of these people will not survive. In the context that doctors at provincial hospitals and health facilities are overloaded, doctors at lower levels have lack of experience, having a tool to support doctors in the process of diagnosing skin diseases quickly and accurately is essential. Along with the strong development of artificial intelligence technologies, many solutions and tools to support the diagnosis of skin diseases have been researched and developed. These include DenseNet, InceptionNet, ResNet, NasNet, SeNet, EfficientNet, VGGNet. In this study, another approach to build tools to aid in the diagnosis of skin pathologies is proposed. SOTA (state of the art) models DenseNet, InceptionNet, ResNet, NasNet, MobileNet combined with Soft-Attention are used as backbone. In addition, personal information such as age and gender are also used. In addition, a new loss function that takes into account the imbalance of the data is also proposed. Experimental results on dataset HAM10000 show that using InceptionResNetV2 with Soft-Attention and new loss function gives 90 percent accuracy, mean of precision, f1-score, recall-score, and auc- scores of 0.81, 0.81, 0.82, and 0.989, respectively, are improvements compared to published indexes. Besides, using MobileNetV3Large combined with Soft-Attention and new loss function, even though the number of parameters is 11 times less, the number of flops is 4 times less, it achieves 86 percent accuracy and 30 times faster diagnosis.

I Introduction

1.1 Problem Statement

Skin cancer is one of the most common cancers leading to worldwide death. Every day, more than 9500[1] people in the United States are diagnosed with skin cancer. Otherwise, 3.6[1] million people are diagnosed with basal cell skin cancer each year. According to the Skin Cancer Foundation, the global incidence of skin cancer continues to increase[2]. In 2019, it is estimated that 192,310 cases of melanoma will be diagnosed in the United States[2]. On the other hand, if patients are early diagnosed, the survival rate is correlated with 99 percent. However, once the disease progresses beyond the skin, survival is poor[2]. Moreover, with the increasing incidence of skin cancers, low awareness among a growing population, and a lack of adequate clinical expertise and services, there is a need for effective solution.

Recently, deep learning particularly, and machine learning in general algorithms have emerged to achieve excellent performance on various tasks, especially in skin disease diagnosis tasks. AI-enabled computer-aided diagnostics (CAD) has solutions in three main categories: Diagnosis, Prognosis, and Medical Treatment. Medical imaging, including ultrasound, computed tomography, magnetic resonance imaging, and X-ray image is used extensively in clinical practice. In Diagnosis, Artificial Intelligence (AI) algorithms are applied for disease detection to save progress execution before these diagnosis results are considered by a doctor. In Prognosis, AI algorithms are used to predict the survival rate of a patient based on his/her history and medical data. In Medical Treatment, AI models are applied to building solutions for a specific disease, medicine revolution is an example. In various studies, AI algorithms have provided various end-to-end solutions in the detection of abnormalities such as breast cancer, brain tumors, lung cancer, esophageal cancer, skin lesions, and foot ulcers across multiple image modalities of medical imaging[2].

In order to adapt the rise in skin cancer case, AI algorithms over the last decade has a great performance. Some typical models that can be mentioned are DenseNet[3], EfficientNet[4], Inception[5], MobileNets[4][6][7], ResNet[8][9], and NasNet[10]. Some of these models which have been used as a backbone model in this paper will be discussed in the Related Work section.

1.2 Literature Review

Skin lesion classification is not a new area, since there are many great performance models constructed. One of the most cutting-edge technologies that have been used is Soft-Attention as stated in [1]. Soumyya and his team construct several models formed by the combination of a backbone model including DenseNet201 [3], InceptionResNetV2 [5], ResNet50 [8] [9], VGG16 [11] and Soft-Attention layer. Using those above backbones has been tried by many previous papers including [12] which uses transfer learning approach with CNN based model, [13] which does not only use those above backbone model but also used InceptionV3 [5] model. Another paper that uses the backbone models is [14]. Hemanth and his team decide to use EfficientNet [15] and SeNET [16] instead and CutOut [17] method which involves creating holes of different sizes on the images i.e. technically making a random portion of image inactive during data augmentation process. [18] also used Deep Convolution Neural Network. Moreover, that paper used RandArgument which crops an image into several images from a fixed size, DropBlock which is used for regularization, Multi-Weighted New Loss which is used for dealing with the imbalanced data problem, end-to-end Cumulative Learning Strategy which can more effectively balance representation learning and classifier learning without additional computational cost. Another state of the art is GradCam and Kernel SHAP [19], which are both model agnostic, local interpretable methods that can highlight pixels that the trained network deems relevant for the final classification.

Otherwise, the Student and Teacher Model is also a state of the art in 2021 [20]. The student and teacher model is the combination of two-mode which share the memory with each other. Therefore, they can take full advantage of what others learn. SkinLinkNet [21] and WonderM [22] are both tested the effect of segmentation on skin lesion classification problem. Another approach is using metadata including gender, age, and capturing position as stated in [23].

On the other hand, skin lesion classification problems are not only applied by Deep Learning but also Machine Learning. Random Forest, XGBoost, and Support Vector Machines are tested by [12]. Besides, Isolation Forest is applied before the soft-max activation of the deep learning

model to detect out of distribution skin lesion images as stated in[13]. Matrix Transformation, besides is also applied before the soft-max activation function in [24].

1.3 Objectives

In this paper, the effect of metadata on classifying skin disease is also analyzed .On the other hand, by analyzing the combination of several backbone models, I will also construct an optimized model that has the ability to classify in a balanced way between classes instead of well identifying the majority of classes.

II Research Result

2.1 Data

2.1.1 Image Data

The dataset used in this paper is the HAM10000 dataset published by Havard University Dataverse[25]. There are total 7 classes in this dataset containing Actinic keratoses and intraepithelial carcinoma or Bowen’s disease (AKIEC), Basal cell Carcinoma (BCC), benign keratosis-like lesions (solar lentigines / seborrheic keratoses andchen-planus like keratoses, BKL), dermatofibroma (DF), melanoma (MEL), melanocytic nevi (NV), and vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, VASC). The distribution of the dataset is shown in the table below:

| Class | AKIEC | BCC | BKL | DF | MEL | NV | VASC | Total |
|------------|-------|-----|------|-----|------|------|------|-------|
| No. Sample | 327 | 514 | 1099 | 115 | 1113 | 6705 | 142 | 10015 |

Table 1: Data Distribution in HAM10000

More than 50 percent of lesions are confirmed through histopathology (HISTO), the ground truth for the rest of the cases is either follow-up examination (FOLLOWUP), expert consensus (CONSENSUS), or confirmation by in-vivo confocal microscopy (CONFOCAL). On the other hand, before being used for training the whole data is shuffled then split into two part. 90 percent and 10 percent of the data is used for training and validating respectively.

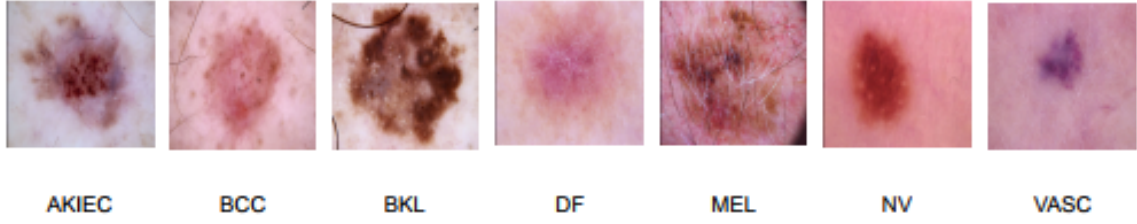


Fig. 1. Example image of each class

In the previous paper[1], the image data is augmented for all class, the number of image increase to 18015 images. Since, this data is imbalanced, using augmented data may cause the problem of well classify on the majority of class. In this paper, instead of augmenting data, metadata is used. The way of processing metadata is discuss in MetaData section. Images in this dataset has the type of *RGB* and shape of (450, 600). However, Each backbone need the different input size of image as well as the range of pixel value. DenseNet201[3] require the input pixels values are scaled between 0 and 1 and each channel is normalized with respect to the ImageNet dataset. In Resnet50 and Resnet152[8][9], the images are converted from *RGB* to *BGR*, then each color channel is zero-centered with respect to the ImageNet dataset, without scaling. InceptionResNetV2[15], on the other hand, will scale input pixels between -1 and 1 . Similarly, three versions of MobileNet[4][6][7], NasNetMobile and NasNetLarge[10] require the input pixel is in range of -1 and 1 .

2.1.2 Metadata

The HAM10000 dataset[25] also contain the metadata of patient including gender, age, and the capturing position. During the data exploration term, the age category miss 57 data point, which is decided to remove this 57 samples. In the gender and capturing position category contain some samples of unknown. Instead of removing, these unknowns data point is kept and considered as "prefer not to say". Besides, the label of the whole data is preprocessed into one-hot vector.

2.2 Model Schema

2.2.1 Input Schema

Using metadata as another input is not new. In paper[23], they decide to keep the missing value and set its value to 0. The sex and anatomical site are categorical encoded. The age, on the other hand is numerical normalized. After processing, the metadata is fed into a two-layer neural network with 256 neurons each. Each layer contains batch normalization, a ReLU[26] activation and dropout with $p = 0.4$. The network's output is concatenated with the CNN's feature vector after global average pooling. Especially, they use a simply data augmentation strategy to address the problem of missing values in metadata. During training, they randomly encode each property as missing with a probability of $p = 0.1$.

In this paper, the unknowns is kept as a type as discussed in Metadata section. Sex, anatomical site and age are also category encoded and numerical normalized, respectively. After processing, the metadata is then concatenated and fed into a dense layer of 4096 neurons. Finally, this this dense layer is then concatenate with the output of Soft-Attention which is then discussed in Soft-Attention section.

The Input schema is described as follow:

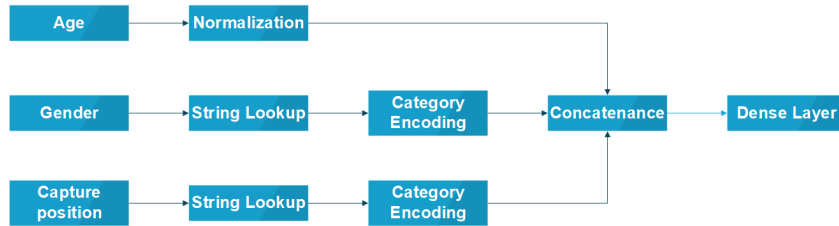


Fig. 2. Input Schema

Image data, on the other hand after being preprocessed, is fed directly into the backbone model.

2.2.2 Soft-Attention

Applying the Soft-Attention layer in deep learning is not a new approach. Soft-Attention has been used in various applications: image caption generation in [27] and handwriting verification in [28] respectively. In skin lesion classification, Soft-Attention is used to increase the performance

of the model as described in [1]. Soft-Attention can ignore irrelevant areas of the image by multiplying the corresponding feature maps with low weights. The function below describes the flow of the Soft-Attention module:

$$f_{sa} = \gamma \sum_{k=1}^K softmax(W_k * t)$$

In order to apply Soft-Attention, there are two main steps. Firstly, the input tensor is put in grid-based feature extraction from the high-resolution image, where each grid cell is analyzed in the whole slide to generate a feature map[29]. This feature map called $t \in R^{h \times w \times d}$ where $h, w, and d$ is the shape of tensor generated by a Convolution Neural Network(CNN), is then input to a 3D convolution layer whose weights is $W_k \in R^{h \times w \times d \times K}$. The output of this convolution is normalized using the softmax function to generate $K = 16$ attention maps. These 16 attention maps are aggregated to produce a weight function called α . This α function is then multiplied with feature tensor t and scaled by γ , a learnable scalar. Finally, the out of Soft-Attention function f_{sa} is the concatenation of the beginning feature tensor t and the scaled attention maps. In this paper, the Soft-Attention layer is applied in the same way in this paper[1]. The

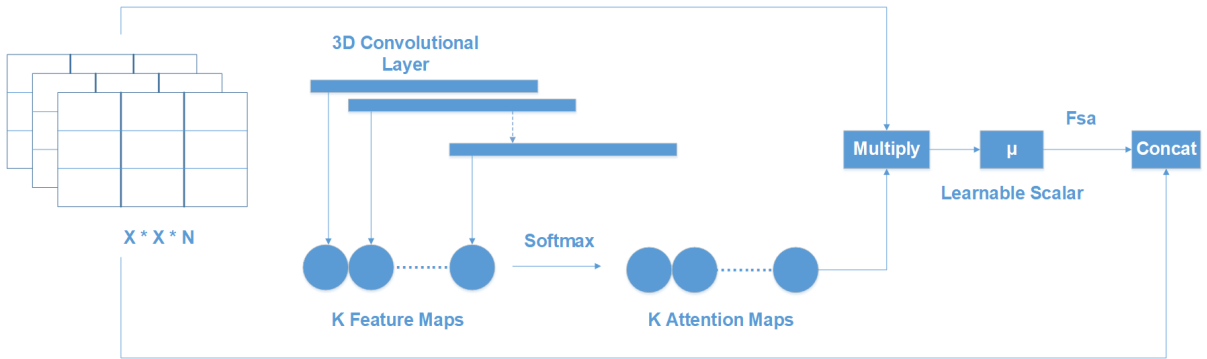


Fig. 3. Soft-Attention Module

Soft-Attention module is described in the following diagram:

2.2.3 Backbone Model Architecture

In this paper, the backbone models that have been used are DenseNet201[3], Inception[5], MobileNets[4][6][7], ResNet[8][9], and NasNet[10]. The combination of DenseNet201, Incep-

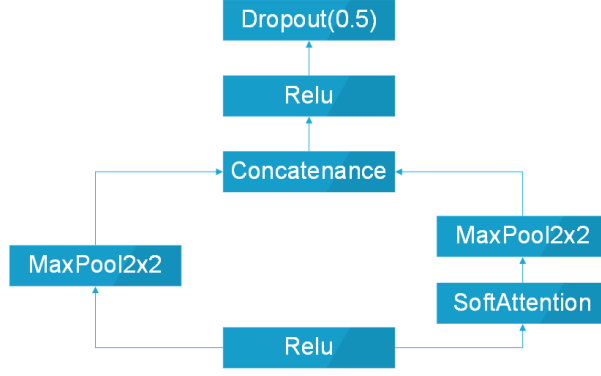


Fig. 4. Soft-Attention Module

tionResNetV2 and Soft-Attention layer are both tested by the previous paper[1] with a great performance. Otherwise, Resnet50 also well classify but with much less number of parameter and depth than based on its f1-score and precision stated. Therefore, in this paper, the performance of the model Resnet152 and NasnetLarge which has the larger number of parameter and depth is analyzed. On the other hand, three version of MobileNet and the NasnetMobile will also be analyzed which has a small number of parameter and depth.

| Model | Size(MB) | Parameters | Depth |
|-------------------|----------|------------|-------|
| Resnet50 | 98 | 25.6M | 107 |
| Resnet152 | 232 | 60.4M | 311 |
| DenseNet201 | 80 | 20.2M | 402 |
| InceptionResNetV2 | 215 | 55.9M | 449 |
| MobileNet | 16 | 4.3M | 55 |
| MobileNetV2 | 14 | 3.5M | 105 |
| MobileNetV3Small | Unknown | 2.5M | 88 |
| MobileNetV3Large | Unknown | 5.5M | 118 |
| NasnetMobile | 23 | 5.3M | 308 |
| NasnetLarge | 343 | 88.9M | 533 |

Table 2: Size and Parameters and Depth of backbone model used in this paper

2.2.4 Model

The whole architecture of the model used for image feature extraction is applied in the same way in paper [1]. Metadata branch, otherwise is preprocessed before feeding into a dense layer then concatenate with the output of Soft-Attention layer. It is described in the figure below:

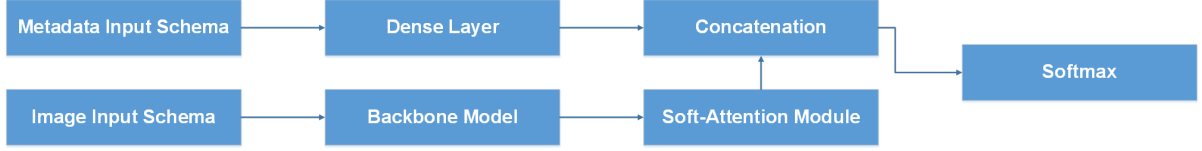


Fig. 5. Overall Model Architecture

2.3 Loss Function

The loss function used in this paper is categorical cross-entropy. Consider $X = [x_1, x_2, \dots, x_n]$ as the input feature, $\theta = [\theta_1, \theta_2, \dots, \theta_n]$. Let N , and C is the number of training examples and number of class respectively. The categorical cross-entropy loss is presented as:

$$L(\theta, x_n) = -\frac{1}{N} \sum_{c=1}^C \sum_{n=1}^N W_c \times y_n^c \times \log(\hat{y}_n^c)$$

where \hat{y}_i^c is the output of model and y_i^c is the target that the model should return, W_c is the weight of class c .

Since the dataset face the imbalanced problem then I applied the class weight for the loss. This formula below is used to calculate the class weight:

$$W = N \odot D$$

$$D = \begin{bmatrix} \frac{1}{C \times N_1} & \frac{1}{C \times N_2} & \dots & \frac{1}{C \times N_n} \end{bmatrix} = \frac{1}{C} \odot \begin{bmatrix} \frac{1}{N_1} & \frac{1}{N_2} & \dots & \frac{1}{N_n} \end{bmatrix}$$

where N is the number of training sample, C is the number of class, N_i is the number of sample in each class i . D is the matrix contain the inverse of $C \times N_i$.

2.4 Evaluation Metrics

In this paper, the model is evaluated by using the confusion matrix and related metrics. The figure 4 illustrates the presentation of a 2×2 confusion matrix used for 2 class. Consider a confusion matrix A with C number of class. Let A^i and A^j is the set of A rows and columns

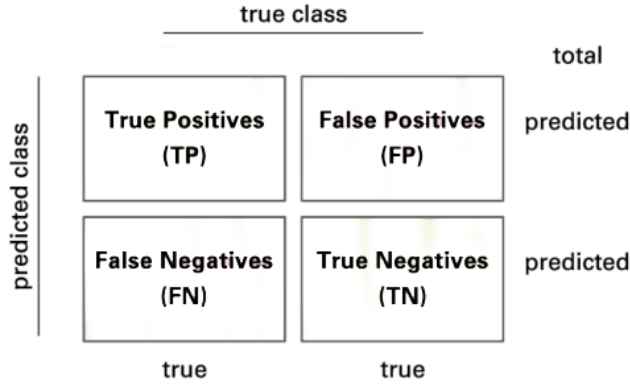


Fig. 6. Confusion Matrix

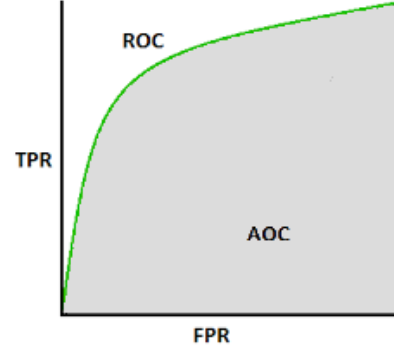


Fig. 7. Area Under the Curve

respectively.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1j} \\ a_{21} & a_{22} & \dots & a_{2j} \\ \vdots & \vdots & & \vdots \\ a_{i1} & a_{i2} & \dots & a_{ij} \end{bmatrix}$$

The True Positive(TP) of all class in this case is the main diagonal of the matrix A. The following method are used to calculate the False Positives(FP), False Negatives(FN), and True Negatives(TN) of all class:

$$FP = -TP + \sum_{k=1}^i A_k^i \quad FN = -TP + \sum_{k=1}^j A_k^j$$

$$TN_c = \sum_{i=1}^C \sum_{j=1}^C a_{ij} - \left[\sum_{k=1}^i A_{i=ck}^i + \sum_{k=1}^j A_{j=ck}^j \right] + a_{i=cj=c} \implies TN = \begin{bmatrix} TN_1 & TN_2 & \dots & TN_c \end{bmatrix}$$

Then, the model is evaluated by the following metrics:

$$\text{Sensitivity(Sens)} = \frac{TP}{TP + FN} \quad \text{Specificity(Spec)} = \frac{TN}{TN + FP}$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad \text{F1 Score} = \frac{2 \times TP}{2 \times TP + FP + FN + TN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN} \quad \text{Balanced Accuracy} = \frac{\text{Sens} + \text{Spec}}{2}$$

The last metric is the *AUC* score standing for Area Under the Curve which is the Receiver Operating Curve(ROC) that indicate the probability of TP versus the probability of FP.

2.5 Training

Before training, the dataset is split into three sub datasets including training, validation, and testing. In this paper, the graphic card that is used to train is Nvidia RTX TitanV.

All the model in this paper is trained with Adam Optimizer[30]. The initial learning rate is set to 0.001, an learning rate reduction schedule is setup with the minimum learning rate is 0.0000001 with the factor of 0.2, the epsilon argument of the optimizer is set to 0.1. On the other hand, Early Stopping is also set up, if the accuracy of validation set dose not increase after 25 epochs, the training will stop.

2.6 Results and Discusstion

The accuracy of all model is presented in the figure below:

| InceptionResNetV2 | DenseNet201 | ResNet50 | VGG16 |
|-------------------|-------------|----------|-------|
| 0.93 | 0.91 | 0.92 | 0.88 |

Table 3: Accuracy of the model with augmented data

| InceptionResNetV2 | DenseNet201 | ResNet50 | Resnet152 | MobileNetV2 |
|-------------------|-------------|----------|-----------|-------------|
| 0.89 | 0.89 | 0.70 | 0.57 | 0.81 |

| MobileNetV3Large | MobileNetV3Small | NasNetLarge | NasNetMobile |
|------------------|------------------|-------------|--------------|
| 0.84 | 0.78 | 0.86 | 0.86 |

Table 4: Accuracy of the model with metadata

According to the Table 3 and 4, it is clear that the model trained with augmented data has a higher accuracy than the model trained with metadata only. While InceptionResNetV2 and DenseNet201 trained with augmented data have accuracy of 0.93 and 0.91 respectively, their training with metadata has both the accuracy of 0.89. Furthermore, Resnet50 trained with augmented data has the accuracy that outperform the Resnet50 and is twice as high as Resnet152 trained with metadata. On the other hand, mobile model including MobileNetV2, MobileNetV3Large, and NasNetMobile, even though has a much smaller number of parameters and depth than the other model, they have a quite good accuracy of 0.81, 0.84, 0.86, respectively. This term will be discussed later on. Since, the model trained with augmented data have high

accuracy, when their f1-score and the recall score of models are analyzed, it turns out that models trained with augmented data have imbalanced f1-scores according to table 5. As a results, augmented data model dose not classify well on all class as InceptionResNetV2 trained on augmented data have 0.29 f1-score on class df while InceptionResNetV2 trained on metadata and the new weight loss can classify well in a balanced way.

| Model | akiec | bcc | bkl | df | mel | nv | vasc | Mean |
|---|-------------|------|-------------|-------------|-------------|------|-------------|-------------|
| DenseNet201 + Augmented Data | 0.67 | 0.78 | 0.64 | 0.67 | 0.61 | 0.96 | 0.91 | 0.74 |
| InceptionResNetV2 + Augmented Data | 0.69 | 0.88 | 0.77 | 0.29 | 0.66 | 0.98 | 1 | 0.75 |
| Resnet50 + Augmented Data | 0.53 | 0.86 | 0.68 | 0.67 | 0.57 | 0.97 | 0.95 | 0.74 |
| VGG16 + Augmented Data | 0.65 | 0.7 | 0.52 | 0.4 | 0.53 | 0.95 | 0.95 | 0.67 |
| DenseNet201 + Metadata and WeightLoss | 0.84 | 0.77 | 0.81 | 0.83 | 0.69 | 0.94 | 0.97 | 0.83 |
| InceptionResNetV2 + Metadata and WeightLoss | 0.77 | 0.83 | 0.83 | 0.64 | 0.75 | 0.94 | 0.7 | 0.81 |
| Resnet50 + Metadata and WeightLoss | 0.49 | 0.59 | 0.55 | 0.36 | 0.45 | 0.83 | 0.8 | 0.58 |
| Resnet152 + Metadata and WeightLoss | 0.42 | 0.38 | 0.41 | 0.15 | 0.4 | 0.75 | 0.75 | 0.46 |
| MobileNetV2 + Metadata and WeightLoss | 0.68 | 0.79 | 0.66 | 0.78 | 0.54 | 0.9 | 0.9 | 0.75 |
| MobileNetV3Large + Metadata and WeightLoss | 0.72 | 0.76 | 0.75 | 0.92 | 0.58 | 0.92 | 0.92 | 0.79 |
| MobileNetV3Small + Metadata and WeightLoss | 0.6 | 0.72 | 0.61 | 0.75 | 0.47 | 0.89 | 0.89 | 0.70 |
| NasNetLarge + Metadata and WeightLoss | 0.79 | 0.79 | 0.8 | 0.74 | 0.65 | 0.92 | 0.92 | 0.80 |
| NasNetMobile + Metadata and WeightLoss | 0.76 | 0.74 | 0.78 | 0.73 | 0.63 | 0.93 | 0.93 | 0.78 |

Table 5: F1-Score of each class and the mean f1-score of each model

However, only DenseNet201, InceptionResNetV2, and NasNetLarge whose depth are equal or larger than 400 have balanced f1-score on class. The others still face the imbalanced term. Since this dataset is not balanced, therefore using augmented data can make the model more bias to the class which has larger sample. Using the metadata, though still make the model bias,

it dose contributes to the improvement of the performance of the model.

| Model | akiec | bcc | bkl | df | mel | nv | vasc | Mean |
|---|-------------|------|-------------|-------------|------|------|----------|-------------|
| DenseNet201 + Augmented Data | 0.52 | 0.77 | 0.58 | 0.5 | 0.71 | 0.97 | 1 | 0.72 |
| InceptionResNetV2 + Augmented Data | 0.52 | 0.88 | 0.83 | 0.17 | 0.65 | 0.98 | 1 | 0.71 |
| Resnet50 + Augmented Data | 0.43 | 0.85 | 0.7 | 0.5 | 0.47 | 0.98 | 0.9 | 0.69 |
| VGG16 + Augmented Data | 0.61 | 0.81 | 0.44 | 0.44 | 0.68 | 0.95 | 0.9 | 0.69 |
| DenseNet201 + Metadata and WeightLoss | 0.85 | 0.75 | 0.78 | 0.83 | 0.63 | 0.96 | 1 | 0.82 |
| InceptionResNetV2 + Metadata and WeightLoss | 0.82 | 0.84 | 0.81 | 0.67 | 0.7 | 0.95 | 0.93 | 0.81 |
| Resnet50 + Metadata and WeightLoss | 0.67 | 0.63 | 0.54 | 0.83 | 0.63 | 0.74 | 0.86 | 0.70 |
| Resnet152 + Metadata and WeightLoss | 0.51 | 0.49 | 0.35 | 0.76 | 0.47 | 0.63 | 0.48 | 0.52 |
| MobileNetV2 + Metadata and WeightLoss | 0.7 | 0.86 | 0.72 | 0.75 | 0.58 | 0.86 | 1 | 0.78 |
| MobileNetV3Large + Metadata and WeightLoss | 0.72 | 0.76 | 0.75 | 0.92 | 0.58 | 0.92 | 0.92 | 0.80 |
| MobileNetV3Small + Metadata and WeightLoss | 0.76 | 0.84 | 0.68 | 1 | 0.52 | 0.82 | 0.93 | 0.79 |
| NasNetLarge + Metadata and WeightLoss | 0.73 | 0.71 | 0.83 | 0.92 | 0.59 | 0.9 | 0.93 | 0.81 |
| NasNetMobile + Metadata and WeightLoss | 0.82 | 0.73 | 0.83 | 0.92 | 0.53 | 0.93 | 0.93 | 0.81 |

Table 6: Recall score of each class and the mean recall score of each model

This problem is also true with the recall score according to table 6. DenseNet201, Inception-ResNetV2, ResNet50, VGG16 trained with augmented data has expected value of recall of 0.72, 0.71, 0.69, 0.69, respectively, while the combination of InceptionResNetV2, Metadata and the new weight loss function achieve the expected value of recall: 0.82. As the results, metadata does improve the model performance by reducing the amount of data needed for achieving nearly the expected results. On the other hand, the reason why the model become much more balanced is the weighted loss function. Weight loss function has ability to solve the imbalanced class samples by adding a weight related to the number of samples in each class. Therefore,

DenseNet201, InceptionResNetV2 trained with the new weighted loss function have recall in akiec of 0.85. 0.82, respectively, as opposed to their training in akiec without weighted loss function. MobileV3large, MobileV3Small, NasNetLarge and NasNetMobile whose standard deviation is less than 0.15 outperform others on classifying class df with the recall score of 0.92, 1, 0.92, 0.92, separately.

Another interesting point found during the experiment is that MobileNetV2, MobileNetV3 and NasNetMobile have small number of parameters and depth, though have relative good performance. The table 8 illustrates the deeper analyzing of the three models. It's clear that MobileNetV3Large and NasNetMobile are the two best performance model.

| Model | [6] | [7]Small | [7]Large | [10]Mobile |
|------------------------|------|----------|-------------|-------------|
| Accuracy(avg) | 0.81 | 0.78 | 0.84 | 0.86 |
| Balanced Accuracy(avg) | 0.86 | 0.87 | 0.87 | 0.88 |
| Precision(avg) | 0.71 | 0.63 | 0.75 | 0.73 |
| F1-score(avg) | 0.75 | 0.70 | 0.79 | 0.78 |
| Sensitivity(avg) | 0.78 | 0.79 | 0.80 | 0.81 |
| Specificity(avg) | 0.95 | 0.95 | 0.95 | 0.96 |
| ROC-AUC-score(avg) | 0.96 | 0.95 | 0.96 | 0.97 |

Table 7: Deeper analyzing of mobile model

Nevertheless, MobileNetV3Large has less number of parameters and depth than NasNetMobile according to the table 2. Therefore, at the end of this experiment, I decide to configure the MobileNetV3Large to construct an optimized model. The result of this experiment is illustrated in table 9.

| Model | Accuracy |
|---|-------------|
| MobileNetV3Large-Layer[:270] + Soft-Attention + Metadata | 0.71 |
| MobileNetV3Large-Layer[:266] + Soft-Attention + Metadata | 0.73 |
| MobileNetV3Large-Layer[:260] + Soft-Attention + Metadata | 0.77 |
| MobileNetV3Large-Layer[:251] + Soft-Attention + Metadata | 0.79 |
| MobileNetV3Large-Layer[:246] + Soft-Attention + Metadata | 0.84 |
| MobileNetV3Large-Layer[:246] + Soft-Attention + Dense(512) + Metadata | 0.85 |
| MobileNetV3Large-Layer[:240] + Soft-Attention + Metadata | 0.86 |
| MobileNetV3Large-Layer[:230] + Soft-Attention + Metadata | 0.84 |
| MobileNetV3Large-Layer[:230] + Soft-Attention | 0.85 |
| MobileNetV3Large | 0.85 |

Table 8: Deeper analyzing of MobileNetV3Large Model

The Table 9 demonstrate that the Soft-Attention and the metadata have a good effect on

the model. Otherwise, Adding a dense layer does not increase the model performance. The best MobileNetV3Large model architecture is the combination of MobileNetV3Large replaced from 241th layer to the rest by Soft-Attention layer and the metadata, which peak the model performance of 0.86 accuracy rate.

| | |
|--------------------|------|
| Accuracy | 0.86 |
| Precision(avg) | 0.86 |
| F1-score(avg) | 0.86 |
| Sensitivity(avg) | 0.86 |
| Specificity(avg) | 0.95 |
| ROC AUC Score(avg) | 0.98 |

Table 9: Performance of MobileNetV3Large

| Model | MobileNetV3Large | DenseNet201 | InceptionResnetV2 |
|---------------------------|------------------|-------------|-------------------|
| No. Parameters | 5.5M | 20.2M | 55.9M |
| Depth | 118 | 402 | 449 |
| Accuracy | 0.86 | 0.89 | 0.90 |
| Time Prediction(s/epochs) | 116 | 1000 | 3500 |

Table 10: How Performance of MobileNetV3Large be optimized

III Conclusion

In this paper, my objective is to analyze the effect of metadata on the performance of model as well as identifying whether metadata can make the model less imbalanced or not. On the other hand, I also try to construct an optimized and balanced model that can be used on mobile phone or electronic devices. The experiment shown that metadata improve the model performance, the factor that make the model much more imbalanced is the augmented data. This problem can be solve quite absolutely by using aforementioned weighted loss function. Using weighted loss function make the expected value of model f1-score increase. At the end of the experiment, mobile model is found that can achieve great performance without either the large number of parameters or depth.

References

- [1] Soumyya Kanti Datta, Mohammad Abuzar Shaikh, Sargur N. Srihari, and Mingchen Gao. Soft-attention improves skin cancer classification performance. Available at <https://arxiv.org/abs/2105.03358>, 4 Jun 2021.
- [2] Manu Goyal, Thomas Knackstedt, Shaofeng Yan, and Saeed Hassanpour. Artificial intelligence-based image classification for diagnosis of skin cancer: Challenges and opportunities. Available at <https://arxiv.org/abs/1911.11872>, 20 Jun 2020.
- [3] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. Available at <https://arxiv.org/abs/1608.06993>, 28 Jan 2018.
- [4] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. Available at <https://arxiv.org/abs/1704.04861>, 17 Apr 2017.
- [5] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. Available at <https://arxiv.org/abs/1512.00567>, 11 Dec 2015.
- [6] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. Available at <https://arxiv.org/abs/1801.04381>, 13 Jan 2018.
- [7] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. Available at <https://arxiv.org/abs/1905.02244>, 20 Nov 2019.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. Available at <https://arxiv.org/abs/1512.03385>, 10 Dec 2015.

- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. Available at <https://arxiv.org/abs/1603.05027>, 25 Jul 2016.
- [10] Jonathon Shlens Quoc V. Le Barret Zoph, Vijay Vasudevan. Learning transferable architectures for scalable image recognition. Available at <https://arxiv.org/abs/1707.07012>, 21 Jul 2017.
- [11] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. Available at <https://arxiv.org/abs/1409.1556>, 10 Apr 2015.
- [12] Rishu Garg, Saumil Maheshwari, and Anupam Shukla. Decision support system for detection and classification of skin cancer using cnn. Available at <https://arxiv.org/abs/1912.03798>, 9 Dec 2019.
- [13] Amirreza Rezvantab, Habib Safigholi, and Somayeh Karimijeshni. Dermatologist level dermoscopy skin cancer classification using different deep learning convolutional neural networks algorithms. Available at <https://arxiv.org/abs/1810.10348>, 21 Oct 2018.
- [14] Hemanth Nadipineni. Method to classify skin lesions using dermoscopic images. Available at <https://arxiv.org/abs/2008.09418>, 21 Aug 2020.
- [15] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. Available at <https://arxiv.org/abs/1905.11946>, 11 Sep 2020.
- [16] Samuel Albanie Gang Sun Enhua Wu Jie Hu, Li Shen. Squeeze-and-excitation networks. Available at <https://arxiv.org/abs/1709.01507>, 16 May 2019.
- [17] Improved Regularization of Convolutional Neural Networks with Cutout. Improved regularization of convolutional neural networks with cutout. Available at <https://arxiv.org/abs/1708.04552v2>, 29 Nov 2017.
- [18] Peng Yao, Mengjuan Xu Shuwei Shen, Peng Liu, Fan Zhang, Jinyu Xing, Pengfei Shao, Benjamin Kaffenberger, and Ronald X. Xu. Single model deep learning on imbalanced small datasets for skin lesion classification. Available at <https://arxiv.org/abs/2102.01284>, 11 Feb 2022.

- [19] Kyle Young, Gareth Booth, Becks Simpson, Reuben Dutton, and Sally Shrapnel. Deep neural network or dermatologist? Available at <https://arxiv.org/abs/1908.06612>, 19 Aug 2019.
- [20] Xiaohan Xing, Yuenan Hou, Hang Li, Yixuan Yuan, Hongsheng Li, and Max Q.-H. Meng. Categorical relation-preserving contrastive knowledge distillation for medical image classification. Available at <https://arxiv.org/abs/2107.03225>, 7 Jul 2021.
- [21] Amirreza Mahbod, Philipp Tschandl, Georg Langs, Rupert Ecker, and Isabella Ellinger. The effects of skin lesion segmentation on the performance of dermoscopic image classification. Available at <https://arxiv.org/abs/2008.12602>, 28 Aug 2020.
- [22] Yeong Chan Lee, Sang-Hyuk Jung, and Hong-Hee Won. Wonderm: Skin lesion classification with fine-tuned neural networks. Available at <https://arxiv.org/abs/1808.03426>, 10 May 2019.
- [23] Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, René Werner, and Alexander Schlaefer. Skin lesion classification using ensembles of multi-resolution efficientnets with meta data. Available at <https://arxiv.org/abs/1910.03910>, 9 Oct 2019.
- [24] Michele Alberti, Angela Botros, Narayan Schuez, Rolf Ingold, Marcus Liwicki, and Mathias Seuret. Trainable spectrally initializable matrix transformations in convolutional neural networks. Available at <https://arxiv.org/abs/1911.05045>, 13 Nov 2019.
- [25] Philipp Tschandl, Cliff Rosendahl, and Harald Kittler. The ham10000 dataset, a large collection of multi-source dermoscopic images of common pigmented skin lesions. Available at <https://arxiv.org/abs/1803.10417>, 25 Nov 2018.
- [26] Abien Fred Agarap. Deep learning using rectified linear units (relu). Available at <https://arxiv.org/abs/1803.08375>, 7 Feb 2019.
- [27] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. Available at <https://arxiv.org/abs/1502.03044>, 19 Apr 2016.

- [28] Mohammad Abuzar Shaikh, Tiehang Duan, Mihir Chauhan, and Sargur N. Srihari. 2020 17th international conference on frontiers in handwriting recognition. Sep 2020.
- [29] Naofumi Tomita, Behnaz Abdollahi, Jason Wei, Bing Ren, Arief Suriawinata, and Saeed Hassanpour. Attention-based deep neural networks for detection of cancerous and precancerous esophagus tissue on histopathological slides. Available at <https://arxiv.org/abs/1811.08513>, 3 Dec 2019.
- [30] Jimmy Ba Diederik P. Kingma. Adam: A method for stochastic optimization. Available at <https://arxiv.org/abs/1412.6980>, 30 Jan 2017.