Deep Learning Exploration on Skin Lesion Data HAM10000

Wei Zhang

Abstract—Skin cancer is the top three most common cancer in the world. It is hard to identify by bare eyes for its similar look on the outside of skins. Previous work had shown promising results in classifying skin cancer with deep learning models. However, the accuracy of identifying the minority class like Melanoma, which has the highest death rate among all skin cancer, is still very low. Therefore, we conducted five data balancing strategies and combined them with deep learning models to improve the minority classification accuracy rate. The dataset HAM10000 was used in the training process, and seven different skin cancer types are classified. The final results showed that the DenseNet201 was outperformed among all the models. The accuracy rate for our DenseNet201 can achieve 92%. Furthermore, the precision for some of the minority classes like Vascular skin lesion can achieve to 100%.

Index Terms—Deep Learning, skin lesions classification, HAM10000, CNN model, transfer learning model

1 Introduction

SKIN cancer is common cancer but can be grown extremely harmful in some particular types. According to WHO, in the three million non-melanoma skin cancer cases, there are about ten thousands cases are melanoma skin cancer [1], which has the highest death rate among all skin cancer. About one-third of cancer over the world is skin cancer, and the most common skin cancer type is Melanocytic nevi. Moreover, it is also difficult to identify different skin cancer types, since many skin lesions look similar (Fig. 1). The misdiagnosis rate of skin cancer is high, and many of the misdiagnoses are life-threatening. The precision of dermatologists to diagnose skin cancer is only 65% to 80% without any technical help [2]. Generally speaking, only the most experienced dermatologists with more than ten years of training can achieve the 80% accuracy rate. With more research on the combination of machine learning and deep learning with cancer problems, the accuracy rate of classifying skin cancer has dramatically improved.

This project aims to examine the existing methods and construct our CNN models to improve the minor classes' precision in the HAM10000 dataset. In section 2, we introduce the previous work on skin data classification. Data description is described in section 3. We implement our method in Section 4 and give results and discussion in section 4. The conclusion is made in section 5.

2 Previous Works

Many efforts for classifying skin cancer have already been made by using machine learning and deep learning method. The accuracy rates can be achieved to 80% to 90% with machine learning studies. Hiam et al.

• *W. Zhang is with the Department of Biostatistics, University of Miami, Miami, FL, 30332.*
E-mail: wxz337@miami.edu

achieved a 92.7% accuracy rate by combining the PCA feature extraction and SVM to classify the lesions [3]. On the deep learning side, Esteva et al. presented a landmark work with a novel tree-based partitioning method that improved the classifying accuracy rate to over 91% [4]. A systematic review by Brinker et al. focused on applying the CNN methods on skin diseases and compared the CNN models' accuracy rate [5]. Although the overall accuracy rate was improved, some malignant types had low accuracy rates, hampered by small sample sizes and imbalanced class problems.

In 2018, the benchmark dataset HAM10000 contained 10015 dermatoscopic images, was released for machine learning and comparisons with human experts [6]. In 2019, Brinker et al. published a paper proved that the CNN model performed better classification results than human beings [7]. Methods like CNN ensemble models with LeNet, ResNets, and VGG architecture could be further improved to better results. However, some skin cancer classes with small sample sizes still do not have an excellent accuracy rate than other large-sized skin cancer types. For example, malignant skin cancer types like Melanoma could only achieve 0.2 to 0.3 precision in many models presented in Kaggle [8]. Therefore, it is crucial to focus on the imbalanced class problem.

3 Data Description

The MNIST HAM10000 dataset was published in 2018. Although the skin datasets before HAM10000 are convincing, most of the dataset showed problem in pathological verification and extremely imbalanced data. Most of the skin cancer images were in the type of nevi or maliganant. The HAM10000 is the most diverse dataset and the sample size of the dataset is large. This dataset was collected from the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia, over a 20-year period [6]. It contained 10015 images of skin pigments with 7 classes: Actinic Keratoses and Intrapithelial Carcinoma (`akiec`), Basal cell carcinoma (`bcc`), benign keratosis (`bkl`), Dermatobroma (`df`), Melanocytic nevi (`nevi`), Melanoma (`mel`), and Vascular skin lesions (`vasc`). The overall size of the data is about 2.7 gigabite. The resolution for each image is 600 $\times$ 450 pixel. We summary the datasets in TABLE 1.

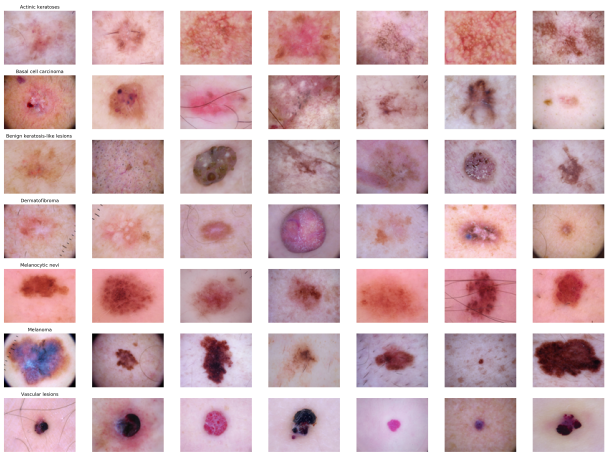


Fig. 1. Image of Seven Types of Skin Lesions

TABLE 1
HAM 10000 Data Descriptions

Features Name	Descriptions	Sample Sizes
akiec	Actinic Keratoses and Intraepithelial Carcinoma	327
bcc	Basal cell carcinoma	514
bkl	"Benign keratosis"	1099
df	Dermatobroma	115
nv	Melanocytic nevi	6705
mel	Melanoma	1113
vasc	Vascular skin lesions	142

4 Methodology

There is an imbalanced class problem with the dataset. Most images are in the `nevi` type. As the Fig 2 shows, the nevi class contains 6,705 images. The smallest class has

only 115 images. Although there are 10,015 images, only 6,705 distinct images. Therefore, we need to handle these problems first.

4.1 Balancing Data Methods

There are two methods to fix imbalanced data: sampling and weights. In the sampling method, two types of sampling are normally used. One is the down-sampling, which samples the majority class to the size of the minority. For the over-sampling method, we sample the minority classes with replacement to a larger size or to the size of the majority classes. In our dataset, the majority class `nevi` contains only 6,705 images, which is about 60 times larger than the smallest minority class. It would lose too much information if we used a down-sampling method. We choose the over-sampling strategy to balanced the data.

Another way is assigning weights to each class. For the majority class, we assign a smaller weight in the loss function. For the minority class, we assign a larger weight. The j_{th} class weight can be calculated as:

$$w_j = \frac{n_j}{j \times n_j}$$

If we using cross-entropy as loss function, The loss function with weight will then becomes:

$$H_y\left(y'\right) = -\sum_i \sum_{j=1}^{K} w_j y_{ij} \log\left(y'_{ij}\right)$$

In this project, we conducted these two methods and fit the models to compare the performance of the models. The following balanced strategies are used in the project:

1) Not adjust for imbalanced data.
2) Adjust with oversampling: only sampling the minority classes with the sizes smaller than 1000.
3) Adjust with oversampling: sampling all minority classes to the sizes of the majority class.

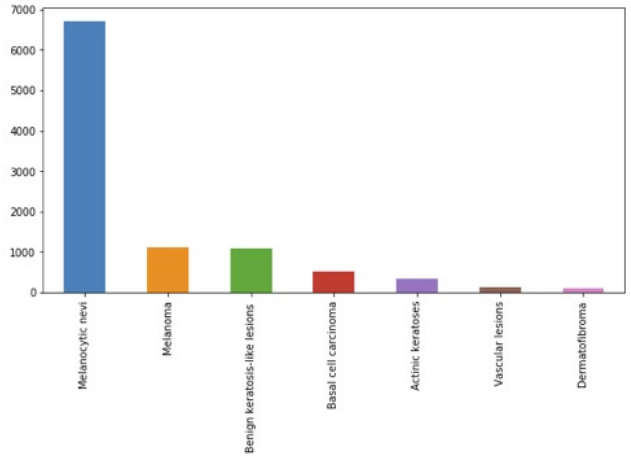


Fig. 2. Imbalanced Data Problem

4) Adjust with weights.
5) Adjust with class weight with oversampling the minority classes with the sizes smaller than 1000.

4.2 Data Preprocessing

The platform we performed our analysis is the Keras TensorFlow in Google Colab. Since the dataset contains only 7,809 non-duplicated images, we used all these image into training set, and seperated the rest of the images into validation set (1,103 images), and test set (1,103 images). We resized the data into 224×224 pixels and converted into float 32. Normalization to $(0, 1)$ scale was applied to the data.

Data augmentation was performed to avoid data overfitting problem. We randomly rotate some of the training images by $90°$, randomly zoom in 10% of some training images, horizontally shift some training images of the width by 10% and vertically shift some training images of the height by 10%.

4.3 Model Building

The model building in the project are mainly Convolutional Neural Network and transfer learning model like ResNet50, GoogleNet InceptionV3, and DenseNet201.

TABLE 2
Architecture of CNN Model

Layers	Output Shape	Parameters
Convolution	(224, 224, 32)	896
Convolution	(224, 224, 32)	9248
Max_Pooling	(112, 112, 32)	0
Dropout(0.25)	(112, 112, 32)	0
Convolution	(112, 112, 64)	18496
Convolution	(112, 112, 64)	36928
Max_Pooling	(56, 56, 64)	0
Dropout(0.4)	(56, 56, 64)	0
Flatten	(200704)	0
Dense	128	25690240
Dropout(0.5)	128	0
Dense	7	903

4.3.1 Convolutional Neural Network

Convolutional Neural Network is assemble by a three-stage neural layers: convolutional layer, pooling, and fully-connected layer. We built our CNN model with the setting in the TABLE 2. Dropout was added after each convolutional layer. The activation function was `relu` in each layers and `softmax` for the output layer. The optimizer we chose was the `Adam` optimizer. For the learning rate, we used an annealing method in other to increase the convergent speed. It was first set with a higher rate and decrease dynamically with every 3 epochs. We use the cross entropy as loss function. The model was performed with 50 epochs and the 16 batch size. Total parameters of the CNN model is 25,756,711.

4.3.2 Transfer Learning Model

For the transfer learning model, we mainly built three models which showed good results in previous work: ResNet50, InceptionV3 and DenseNet201.

- **ResNet50** is a deep residual network with 50 layers in it. Skipping connection is the main innovation of ResNet. Deep networks often suffer from vanishing gradients without adjustments. However, the skip connection allows the network to learn its identity function. The network could pass the input without go through the other weight layers. The vaninshing gradient is then offsetting with it [9].
- **InceptionV3** is a convolutional neural network that focus on using less computational power. The architecture for an InceptionV3 is: factorized convolutions, smaller convoluntions, asymmetric convolutions, auxiliary classifier, and grid size reduction [10].
- **DenseNet201** is a dense convolutional network which connects each layer to every other layer in a feed-forward way with 201 layers. The DenseNets obtained significant improvements in alleviating the gradient vanishing problem and substantially reducing the number of parameters [11].

In the project, all the transfer learning models are pre-trained on the ImageNet. We modified our transfer learning model as following: 1. Instead of using the average pooling, we use the global average pooling. 2. We removed the top layer but added a flatten layer, a fully connected layer, a dropout layer and batch normalization (Fig. 3). TABLE 3 is an example of our architecture of ResNet50.

The total parameter for the ResNet50 model is 25,697,159. For the InceptionV3 model, the total parameter is 23,875,751. And the DenseNet201 has only 20,300,359 total parameter, which is the smallest parameter number in all the model we built in this project. The optimizer we chose was the `RMSprop` with a small learning rate to avoid overfitting problem. The loss function was the cross entropy. The model was performed with 30 epochs and the 16 batch size.

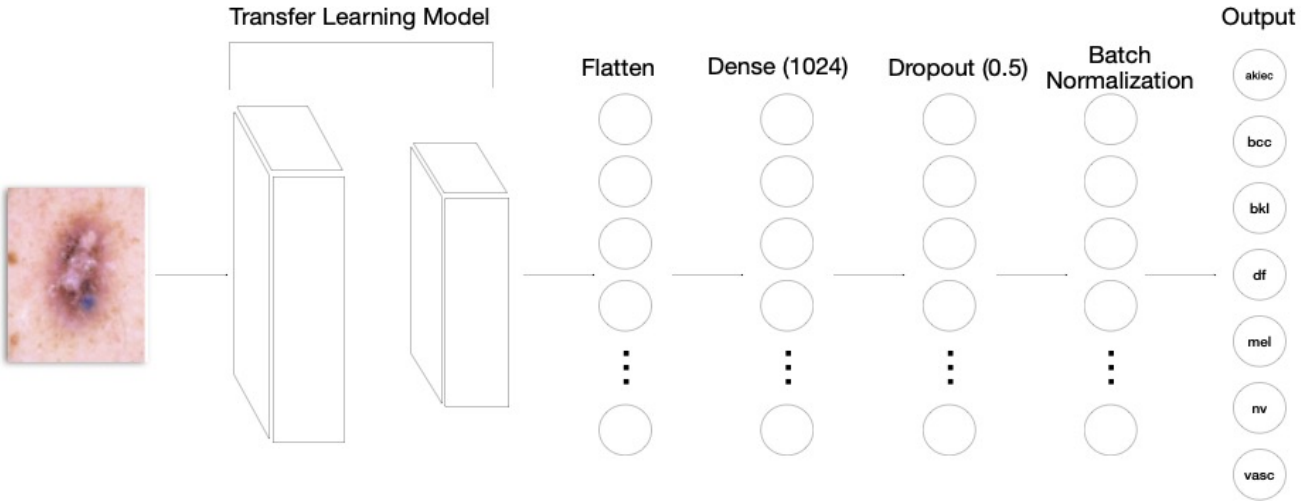


Fig. 3. Architecture of Transfer Learning

TABLE 3
Architecture of ResNet50 Model

Layers	Output Shape	Parameters
ResNet50	2048	23587712
Flatten	2048	0
Dense	1024	2098176
Dropout(0.5)	1024	0
BatchNormalization	1024	4096
Dense	7	7175

5 Result and Discussion

5.1 Evaluation Criterion

The following criterions will use in our project to evaluate the performance of the models:

Accuracy: True predictions over all predictions.

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

Precision: True positive prediction over all positive prediction.

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

Recall: True positive prediction over all positive samples.

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

F1-score: Balanced precision and recall.

$$\text{F1 - score} = \frac{2TP}{2TP + FP + FN}$$

5.2 Result

5.2.1 Results Between Balanced Strategies

We first compared the results between different balancing data strategy. The test accuracy for different models shows in TABLE 4. DenseNet201 shows the best test accuracy of 0.921 among all models. Moreover, strategy 5, with both oversampling and weights, gives us the best performance. The results between adding weights and two types of oversampling are similar. In the CNN models, the gap of test accuracy between different strategies is large. Adding weights for the CNN model is the worst scenario. For the ResNet50, augment sampling or adding weights give us similar results in test accuracy. Not adding the oversampling or weights are slightly lower the test accuracy. However, the gap is very tiny. The

TABLE 4
Test Accuracy for Different Strategies and models

Strategy/Model	CNN	ResNet50	InceptionV3	DenseNet201
S1	0.789	0.889	**0.899**	0.889
S2	**0.844**	0.894	0.887	0.909
S3	**0.844**	**0.906**	0.897	0.915
S4	0.031	**0.905**	**0.899**	0.910
S5	0.543	0.902	0.898	**0.921**

TABLE 5
Precision

Model	akiec	bcc	bkl	df	nv	mel	vasc	Overall
CNN	**1**	0.53	0.48	0.00	0.87	0.36	0.83	0.84
ResNet50	0.42	0.74	0.70	**0.67**	0.96	0.58	0.93	0.90
InceptionV3	0.69	**0.84**	0.63	0.62	**0.97**	0.49	0.69	0.90
DenseNet201	0.74	0.81	**0.82**	0.60	0.94	**0.73**	**1**	**0.92**

InceptionV3 shows no difference in all five strategies.

5.2.2 *Results Between Models*

In each model, we select the best test accuracy in all strategies and compare the precision (TABLE 5), recall (TABLE 6), and F1-score (TABLE 7) for each class.

For the smallest minority class `df`, although the CNN model has a good overall accuracy of 0.84, the minority class prediction is much worse than other models. As the scores showed in all tables and the confusion matrix (Fig.4) of the CNN model, we can see that no images were classified to the `df` class. ResNet50 gives us the best precision and F1-score, and InceptionV3 gives us the best recall score of 0.80 for class `df`.

For the second small class `vasc`, DenseNet201 outperformed among all models in terms of all evaluation criteria. For the largest class textttnv, all models performed very well. DenseNet201 gives us the best overall performance (See Confusion Matrix Fig 5.). Moreover, for the malignant skin cancer (`mel`), the DenseNet achieved 0.73 in precision score. These results are much improved compared with our baseline model.

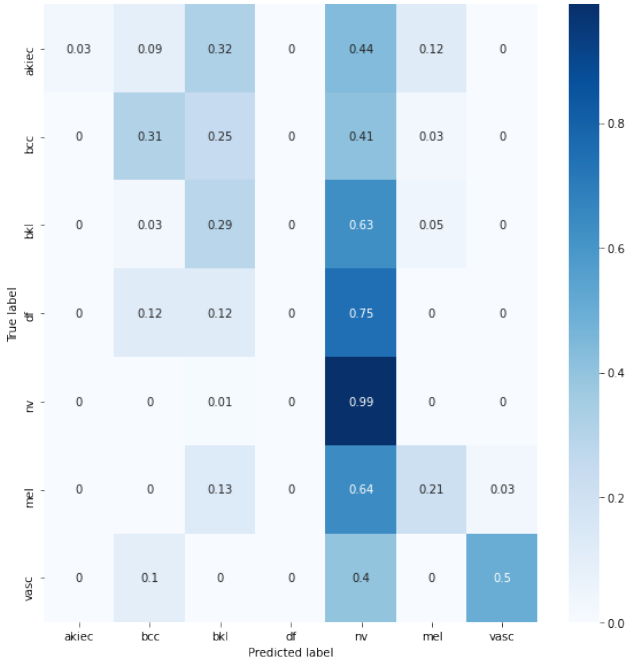


Fig. 4. Best CNN Model Confusion Matrix

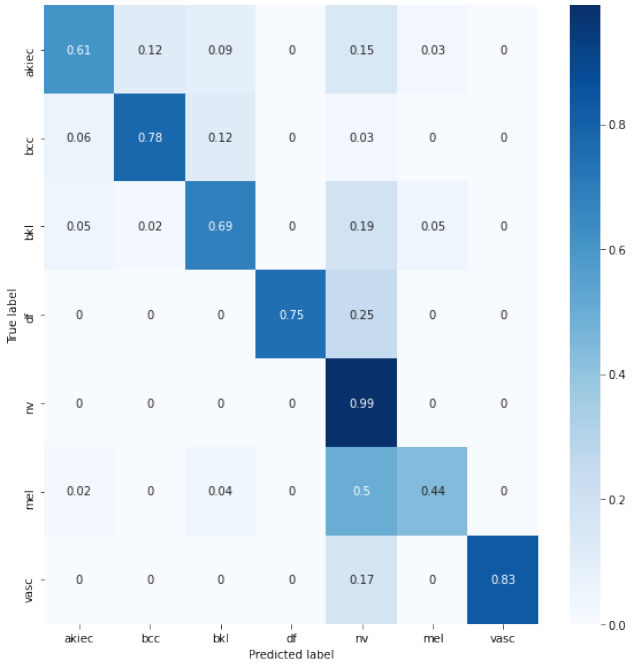
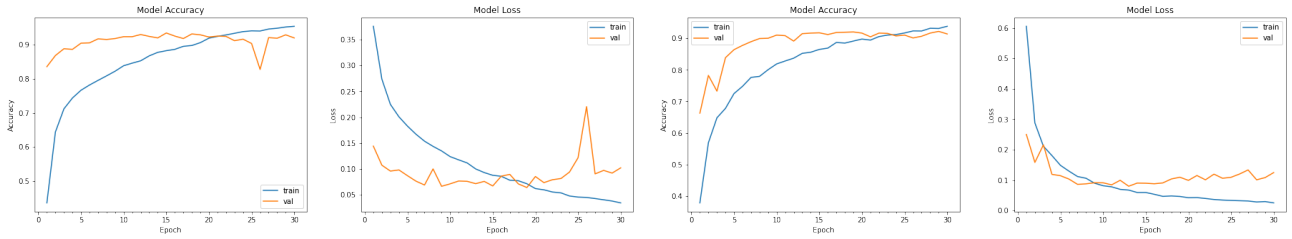


Fig. 5. Best DenseNet201 Confusion Matrix

5.3 Discussion

For different models, using a suitable balancing data method is important. Take the CNN model as an example. If we used the



(a) DenseNet201 with Strategy 3 (b) DenseNet201 with Strategy 5

Fig. 6. Simulation results.

TABLE 6
Recall

Model	akiec	bcc	bkl	df	nv	mel	vasc	Overall
CNN	0.03	0.31	0.29	0.00	**0.99**	0.21	0.50	0.84
ResNet50	0.42	**0.78**	0.65	0.75	0.98	0.39	0.81	0.90
InceptionV3	0.56	0.77	**0.72**	**0.80**	0.95	**0.59**	0.69	0.90
DenseNet201	**0.61**	**0.78**	0.69	0.75	0.99	0.44	**0.83**	**0.92**

TABLE 7
F1-score

Model	akiec	bcc	bkl	df	nv	mel	vasc	Overall
CNN	0.06	0.39	0.36	0.00	0.93	0.26	0.62	0.84
ResNet50	0.42	0.76	0.67	**0.71**	**0.97**	0.47	0.87	0.90
InceptionV3	0.62	0.81	0.67	0.70	0.96	0.54	0.69	0.90
DenseNet201	**0.67**	**0.79**	**0.75**	0.67	**0.97**	**0.55**	**0.91**	0.92

oversampling strategy, the result in terms of test accuracy was better than not using any balancing data method. However, if we added weights to the CNN model, the result is much worse than expected. Moreover, in the transfer learning model, the model with weights is more stable than the model with data oversampling. As we can see in the model performance over epochs (Fig. 6), the model with oversampling (Fig. 6a) was jumped in between epoch 30 and 50. The model with both oversampling and weights did not have this jump (Fig. 6b).

6 Conclusion

In this project, we trained end-to-end deep learning model without feature selection or preprocessing. For the final classification results of the HAM10000, transfer learning models were beated the CNN models. By reusing and modifying pre-trained model architecture, and applying the balancing data strategy, the DenseNet201 achieved 92% accuracy rate. The minority class accuracy had also gained a good improvement. Moreover, we compared the impact of different balancing data strategy to the models. The overall performance is better when balancing the data. For the future work, we will implement other model structures and focus on other methods dealing with the imbalanced data problem.

References

[1] World Health Organization, Projections of mortality and causes of death, 2016 to 2060, *Health statistics and information system*. Available: https://www.who.int/healthinfo/globalburdendisease/projections/en/.

[2] Argenziano, G., & Soyer, H. P. Dermoscopy of pigmented skin lesions–a valuable tool for early diagnosis of melanoma. The Lancet. Oncology, 2(7), 443–449(2001). https://doi.org/10.1016/s1470-2045(00)00422-8

[3] Alquran,H., et al. The melanoma skin cancer detection and classification using support vector machine. *2017 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT)*, Aqaba, 2017, pp. 1-5. doi: 10.1109/AEECT.2017.8257738

[4] Esteva, A., Kuprel, B., Novoa, R. et al. Dermatologist-level classification of skin cancer with deep neural networks. *Nature* 542, 115–118 (2017). https://doi-org.access.library.miami.edu/10.1038/nature21056

[5] Brinker, T. J., Hekler, A., Utikal, J. S. et al. Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review. *Journal of medical Internet research*, 20(10), e11936 (2018). https://doi.org/10.2196/11936

[6] Tschandl, P., Rosendahl, C. & Kittler, H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. *Sci. Data 5*, 180161 (2018). doi:10.1038/sdata.2018.161

[7] Brinker, T. J. et al. & Poetri, R. *A convolutional neural network trained with dermoscopic images performed on par with 145 dermatologists in a clinical melanoma image classification task.* Eur. J. Cancer. 2019, doi: 10.1016/j.ejca.2019.02.005.

[8] Mader,K. Skin Cancer MNIST: HAM10000. *Kaggle*.

[9] He, K., et al. Deep Residual Learning for Image Recognition, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.
Available: https://www.kaggle.com/kmader/skincancer-mnist-ham10000

[10] Kurama, V. A Guide to ResNet, Inception v3, and SqueezeNet. *Paperspace Blog*, 8 June 2020,
Available: blog.paperspace.com/popular-deep-learning-architectures-resnet-inceptionv3-squeezenet/.

[11] Huang, G et al., Densely Connected Convolutional Networks, *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017, pp. 2261-2269, doi: 10.1109/CVPR.2017.243.