Capstone Project

Hsin-Wei Wang
25 August 2019

Definition

Project Overview

Skin cancer is one of the most common cancers worldwide, and it has two main types: melanoma and non-melanoma. Non-melanoma skin cancer is the fifth most common cancer with more than 1 million diagnoses worldwide in 2018. Basal cell carcinoma and squamous cell carcinoma are the most common categories of non-melanoma tumors [1]. Melanoma is much more lethal than non-melanoma. It accounts for less than 5% of skin cancers in the United States but causes about 75% skin-cancer-related deaths [2]. The incidence rate of skin cancer is increasing worldwide and has become a significant economic burden for public health services [3]. From 2006 to 2015, the incidence rate of melanoma increased by 3% per year for people ages 50 and older but was stable for those younger than 50. The new cases and deaths of melanoma in the US in 2019 are estimated at 96,480 and 7,230 respectively [4]. Both melanoma and non-melanoma cancers are highly curable if the cancer is diagnosed and treated in its early stage. The estimated 5-year relative survival rate for patients whose melanoma is detected early is about 98%, but the survival rate drops to 23% if malignant cancer metastasizes to parts of the body remote from the primary tumor [4]. Therefore, a fast, automated and accurate diagnosis may help dermatologists and patients detect skin cancer earlier and lead to a high chance of survival.

To facilitate the application of skin images to help reduce skin cancer mortality, the International Skin Imaging Collaboration (ISIC) has created the ISIC Archive [5-8], the largest publicly available collection of quality controlled dermoscopic images of skin lesions as a benchmark for education and research. Since 2016, they have organized the annual "ISIC: Skin Lesion Analysis Towards Melanoma Detection" challenge including problems such as lesion segmentation, lesion attribute detection, and disease classification. These challenges have attracted global participation and encouraged novel automated algorithms.

Problem Statement

The ISIC 2019 challenge contains two tasks: classify dermoscopic images without meta-data and classify images with additional meta-data [9]. In this project, I will only focus on Task 1. The goal of both tasks is to classify 8,238 dermoscopic images of the test data into 9 different diagnostic categories as shown in Table 1. There are 25,331 images unequally distributed across 8 different categories in the training data, but the test data contains an additional outlier category not presented in the training data. Therefore, the proposed algorithm should be able to identify the additional category. This task is more difficult than the classification task in ISIC 2018 challenge which only has 7 possible disease categories and no need to tackle with novelty detection problem [10]. The submission file to the challenge is a CSV (comma-separated value) file which contains predicted diagnosis confidences of the 9 categories for each image. A public leaderboard and final scores will be released on 30 August 2019.

I have submitted two prediction results according to two slightly different approaches. The approach-1 consists of an ensemble of convolutional neural networks (CNNs) for classifying 8 known categories and an out-of-distribution detector to determine whether an image belongs to the

unknown category. The approach-2 is much more naive than the approach-1 and does not apply any out-of-distribution detector. Instead, it directly uses an ensemble of CNNs to classify the 9 categories. The details of the approaches are introduced in the Methodology section.

Table 1. The number of images for each category in the training dataset

Diagnostic Category	Amount	Percentage
Melanoma (MEL)	4522	17.85
Melanocytic nevus (NV)	12875	50.83
Basal cell carcinoma (BCC)	3323	13.12
Actinic keratosis (AK)	867	3.42
Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis) (BKL)	2624	10.36
Dermatofibroma (DF)	239	0.94
Vascular lesion (VASC)	253	1.00
Squamous cell carcinoma (SCC)	628	2.48
None of the others (UNK)	0	0

Metrics

One main goal of clinical application of automatic skin lesion classification is acquiring specific information and treatment options for a lesion, and achieving this goal depends on correct diagnosis out of multiple categories. In previous challenges, evaluation criteria were melanoma average precision and melanoma AUROC (area under the receiver operating characteristic curve), but they are only robust to the prevalence imbalance of melanomas and could be influenced by clinically irrelevant low-recall regions [11]. Balanced multi-class accuracy (BMCA) is helpful to select a classifier which is not over-fitting to arbitrary dataset imbalance, as is often the case with accuracy. Therefore, the goal metric became BMCA since ISIC 2018 to minimize the influence of dataset prevalence and distribution that may not be consistent with the real-world situation. AUROC was the second metric to break tied positions. BMCA calculates accuracy on a per-class basis and then averages those accuracies as defined:

$$BMCA = \text{mean}\left(\sum_{i=1}^{C} \frac{TP_i}{TP_i + FN_i}\right) \quad (1)$$

where C denotes the number of diagnostic categories, TP_i and FN_i are the number of true positives and false negatives for the ith category respectively. The best value of BMCA is 1 that represents all samples are classified correctly. Conversely, the worst value of BMCA is 0 when all samples are misclassified.

Analysis

Data Exploration and Visualization

The dermoscopic images provided by the ISIC organization are compiled from a variety of devices within internationally leading clinical centers to ensure clinically representative. These images are in 24-bit (8 bits per channel) JPEG format and EXIF tags within the images have been removed. Dimensions of these images are inconsistent ranging from 600 × 450 to 1024 × 1024. Figure 1 shows a few samples for each category in the training data. It's hard for a general person to visually identify consensus features within a category or the properties that differentiate different types of skin lesions.

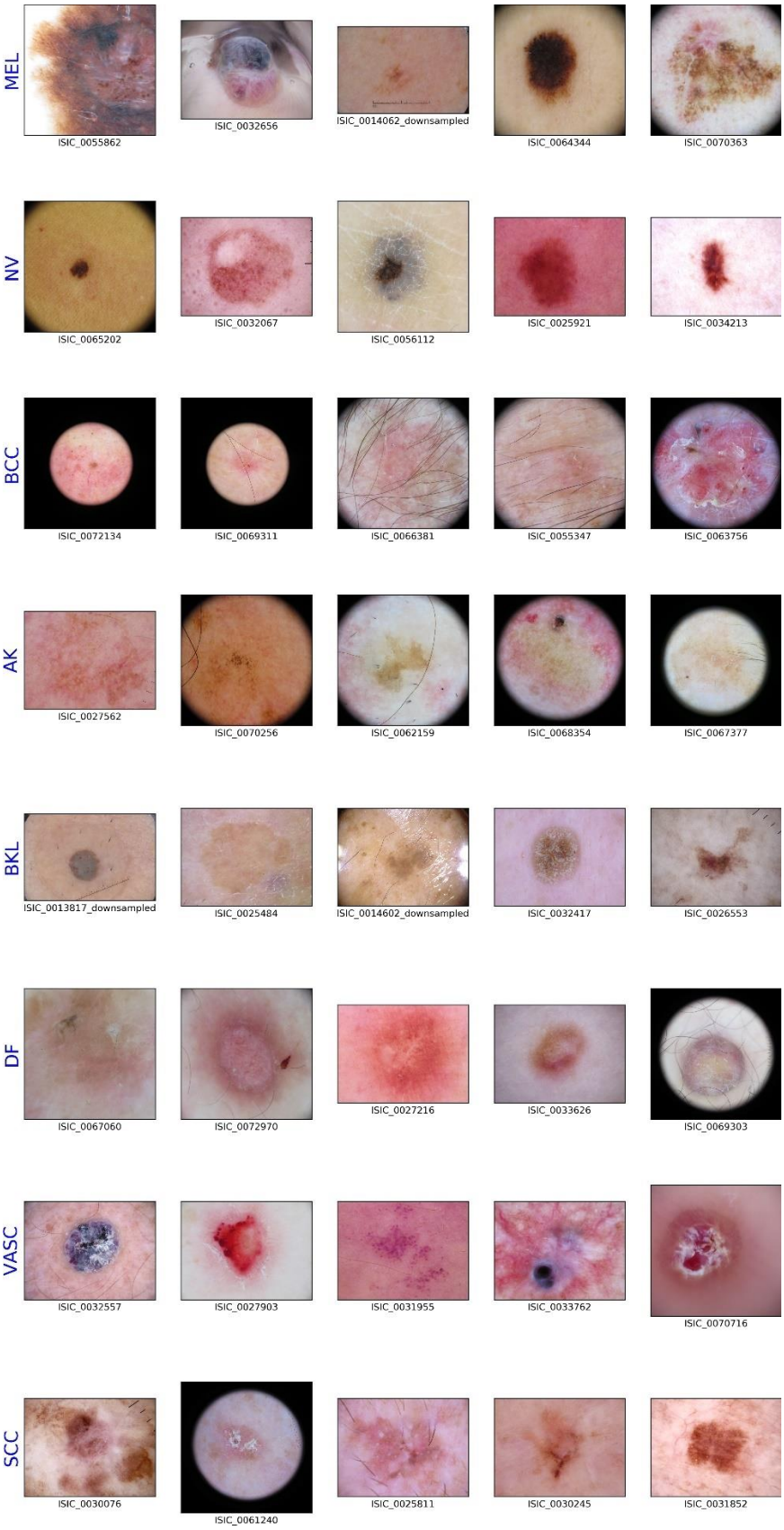


Figure 1 Samples of the training data among 8 categories

Besides the training data, I have collected an out-of-distribution dataset containing 134 dermoscopic images as shown in Table 2. 118 images of 6 various categories and 16 melanosis images were compiled from ISIC Archive and 7-Point dataset [12] respectively. Each out-of-distribution image was pairwise compared with all images in the training dataset by perceptual hashing algorithm followed by manually visual comparison, and no duplicate skin lesions have been found between two datasets. This was done in the Jupyter Notebook *find_duplicates.ipynb*.

Table 2. The number of images for each category in the out-of-distribution dataset

Diagnostic Category	Amount	Source
Angiofibroma or fibrous papule	1	ISIC Archive
Angioma	12	
Atypical melanocytic proliferation	12	
Lentigo NOS	70	
Lentigo simplex	22	
Scar	1	
Melanosis	16	7-Point

Approach-1 and 2 used the datasets in different ways. For approach-1, the original training data were randomly split into 80% training set (20264 images) and 20% validation set (5067 images) over each category. The training set was used to train three deep neural networks for classifying images into one of the 8 disclosed categories, and the out-of-distribution dataset was for tuning the parameters of an out-of-distribution detector. For approach-2, two duplicate and mislabeled images (ISIC_0069013, ISIC_0067980) later found in the training data were removed. They were not removed in approach-1 simply due to no more time and budget to re-train the models. All images of the out-of-distribution dataset were regarded as the unknown category. Both datasets were together randomly split into 80% training set (20370 images) and 20% validation set (5093 images) over each category. The training set was used to train three deep neural networks for classifying images into one of the 9 categories.

Algorithms and Techniques

Previous studies have shown that skin lesion classifiers based on CNNs achieved performance on par with dermatologists [13]. Presented CNN methods that use transfer learning and parameters fine-tuning for skin lesion classification are the most common and best-performing approaches. I also adopted a similar approach using transfer learning to train CNN models. CNN is a type of deep neural network that employs convolution in at least one of its layers [14]. It has led breakthroughs in processing image, video, speech, audio, and become the dominant machine learning approach for visual recognition [15, 16]. Unlike multilayer perceptrons (MLPs) which only accept vectors as input and only use fully connected (FC) layers, CNNs also accept matrices as input and use sparsely connected layers. A CNN comprises an input layer, multiple hidden layers, and an output layer. The input layer usually takes matrices with fixed shape, hence an image is resized to the target shape before inputted to the CNN. The hidden layers are typically composed of a cascade of interlaced convolutional layers and pooling layers for feature extraction and transformation. This sequence of layers discovers spatial patterns contained in the image and gradually learns to recognize basic lines, curves, shapes, blobs, and increasingly complex objects. These features are fed into the output layer which consists of one or more FC layers to identify the final object within the input image. We don't have to specify or program the CNN to recognize any specific features, instead, CNN learns all of these on its own from training data through the backpropagation algorithm.

Building a successful CNN model from scratch demands lots of works on designing architectures and training them on large datasets iteratively. Nowadays, there are several public state-of-the-art models which were previously trained on a large image dataset containing more than one million images, e.g. ImageNet [17], for large-scale classification task. The idea behind transfer learning is that instead of constructing a CNN from scratch, how do we leverage the knowledge learned in these pre-trained models to train a new model for another problem? Since the pre-trained models were trained on a significantly large and general dataset, they have learned certain common low-level features such as edges, corners, blobs, and stripes. These basic features can be shared across different visual recognition tasks, and thus enable transferring the knowledge of the pre-trained model to a new one. Transfer learning involves taking a pre-trained neural network and adapting it to a new dataset. The approach for using transfer learning will be different depending on the size of the new dataset, and the similarity between the original and new datasets. Here I followed the previously successful approach for skin lesion classification, and the implementation details of transfer learning are described in the CNN Training section.

Benchmark

There were 139 algorithms created by 77 machine-learning labs participating in the ISIC 2018 [18]. The ISIC 2019 challenge is an open and ongoing contest. Each algorithm submitted to this challenge can be regarded as a benchmark model. Since there won't be any published results available for comparison until 30 August 2019, I have created a simple vanilla CNN as benchmark model which has 4 convolutional layers, and each is followed by a max-pooling layer. A dropout layer and a global average pooling layer are added after the last pooling layer and followed by a dense layer and a softmax activation layer. The input images were resized to 224 × 224 pixels to be consistent with most ImageNet pre-trained models. The vanilla CNN was only compared with the models of approach-1 and not further integrated with the ODIN method since it performed significantly worse than the other models.

Methodology

Data Preprocessing

The input to the CNN was processed images and diagnostic category labels. In order to prevent overfitting and make the model generalize better, image augmentation was applied during training. First, images were cropped by an arbitrary percentage between 0.8 and 1 of their area. Next, random rotation of 90, 180, or 270 degrees with a probability of 0.5 was performed. Then I applied random vertical flip, random horizontal flip, random brightness change and random saturation change in sequence with a probability of 0.5 for each operation. Furthermore, images were resized to the input size of the model as shown in Table 3. Finally, pixels were scaled between -1 and 1 for Xception. For DenseNet201 and ResNeXt50, images were normalized by the per-channel mean and standard deviation which were calculated over the training set.

Implementation and Refinement

This project was implemented by Keras with TensorFlow backend. The source codes are available at https://github.com/wanghsinwei/isic-2019. To reproduce the training and predicting processes of both approaches, just follow the descriptions of the readme.

Image Augmentation

Some functions such as cropping and saturation change are not provided by the *ImageDataGenerator* class of Keras, hence I utilized the Augmentor package by forking it and made a few modifications. The *ImageIterator* class defined in *image_iterator.py* was designed to integrate the augmentation pipeline of Augmentor with the training processes of Keras.

Balanced Multi-class Accuracy

Keras only supports a few metrics and BMCA is not included. In the beginning, I thought it's okay to directly call tf.metrics.mean_per_class_accuracy of TensorFlow in the custom metric function. However, I found the calculated accuracies were incorrect when checking the complexity graphs of training processes. Finally, I defined a custom function in *metrics.py* to calculate BMCA during training.

CNN Training

Three state-of-the-art architectures listed in Table 3 were used for transfer learning. These models were pre-trained on the ImageNet dataset and then fine-tuned on the training set. In the proposal, I was planning to adopt NASNet-Large architecture, but I found this model with default input size is too large to be trained using a single 16 GB graphic card unless the batch size is very small. Therefore, it was replaced by ResNeXt50. All models followed the same setup. The original top FC layer was replaced by new layers including one FC layer with 512 units, one dropout layer with 0.3 dropout rate, and one FC layer with 8 or 9 units for approach-1 or 2 respectively followed by one softmax activation layer. The ensemble model was simply created by taking the average of softmax probabilities made by the three models. These average probabilities are the predicted diagnosis confidences for each category.

Table 3. CNN architectures for transfer learning

Architecture	Input Size	Parameters
DenseNet201	224 × 224	20.2 M
Xception	299 × 299	22.9 M
ResNeXt50	224 × 224	25.1 M

The class weighted cross-entropy loss was applied during training to mitigate the data imbalance problem. The weight of each category is defined as

$$w_i = \frac{N}{C \times n_i} \quad (2)$$

where N denotes the total number of samples in the training set, n_i is the number of samples for category i, and C is the number of categories.

All layers of base models were frozen first to perform feature extraction, so the weights of newly added layers are not completely random. This process is critical and can prevent the magnitude of the gradient updates being too large when fine-tuning. Without this process, the models converged slower and the losses were much higher. After 3 epochs of feature extraction, all layers were unfrozen to fine-tune the models. The 4th-rank method in the ISIC 2018 challenge have tested Adam, Nadam, and RMSprop optimizers and found that Adam generally performed best [19]. Therefore, all three transfer learning models were trained using Adam optimizer. Each model was trained with a batch size of 32, and the learning rate was set as 10^{-4} for feature extraction. Then in the fine-tuning step, the starting learning rate was set as 10^{-5} and reduced by a factor of 10 if the validation loss has stopped improving for 8 epochs until it reaches 10^{-7}. The early stopping

strategy monitoring the validation loss with the patience of 16 epochs was also applied. The model which attained the best BMCA on the validation set during training will be kept as the final model. The vanilla CNN followed a similar setup except it doesn't have the fine-tuning step, and its starting learning rate was set as 10^{-3} and minimum learning rate was 10^{-6}.

Out-of-distribution Detector

An intuitive solution for detecting an additional outlier class on the test dataset might be to expand the training dataset by collecting or generating out-of-distribution samples. However, the scope of the outlier is not specified clearly, therefore the number of possible out-of-distribution samples can be extremely large and difficult to compile. To tackle this problem, I leveraged a simple and easily implementable method, ODIN (**O**ut-of-**DI**stribution detector for **N**eural networks) [20], for distinguishing in- and out-of-distribution samples in approach-1. ODIN is based on two techniques, temperature scaling and input perturbation. This method does not require retraining the neural network and has been proven effective on several different network architectures. Hence, it can be easily integrated with the models trained using the above steps. ODIN seems originally implemented by PyTorch v0.3.1. In order to re-implement it through Keras, I tried to run their codes on Google Colab with some code changes (see commits and notebook). Most of my implementations about ODIN are in *odin.py*.

There are 3 parameters of ODIN, T (temperature scaling), ε (perturbation magnitude) and δ (threshold) need tuning. The images of the validation set and the out-of-distribution dataset were regarded as in- and out-distribution samples respectively. Random 50% of both samples were used to tune the parameters and the rest for testing. I followed the same parameter searching scopes in [20] to choose the combination of T, ε, and δ that achieved the minimum of false positive rate (FPR) when the true positive rate (TPR) was 95%. T was selected among 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000; and ε was chosen from 21 evenly spaced numbers in [0, 0.004].

The softmax score of ODIN was used to determine the diagnosis confidence of the unknown category. Instead of just giving confidence of 0 or 1 based on δ, the softmax score was rescaled using the logistic function:

$$\frac{1}{1 + e^{-k(s-\delta)}} \quad (3)$$

where s denotes the softmax score of a sample and k is the logistic growth rate which determines the steepness of the curve. The confidence is 0.5 when the softmax score equals to the threshold δ. k was chosen from one of the 1000 integers in [1, 1000] which obtained the highest BMCA of 9 categories. All images of the out-of-distribution dataset were regarded as the unknown category.

Results

Model Evaluation and Validation

8-category Classification of Approach-1

The evaluation results of CNN models on the validation set are shown in Table 4. As expected, the vanilla model performed much worse than the other models in terms of both loss and BMCA. The ensemble of CNNs outperformed each individual model and achieved a BMCA of 0.838. Figure 2 shows the confusion matrixes of all models over the 8 categories on the validation set. The distributions of misclassification are similar among these models excluding Vanilla, and the SCC category has the largest portion of images being misclassified.

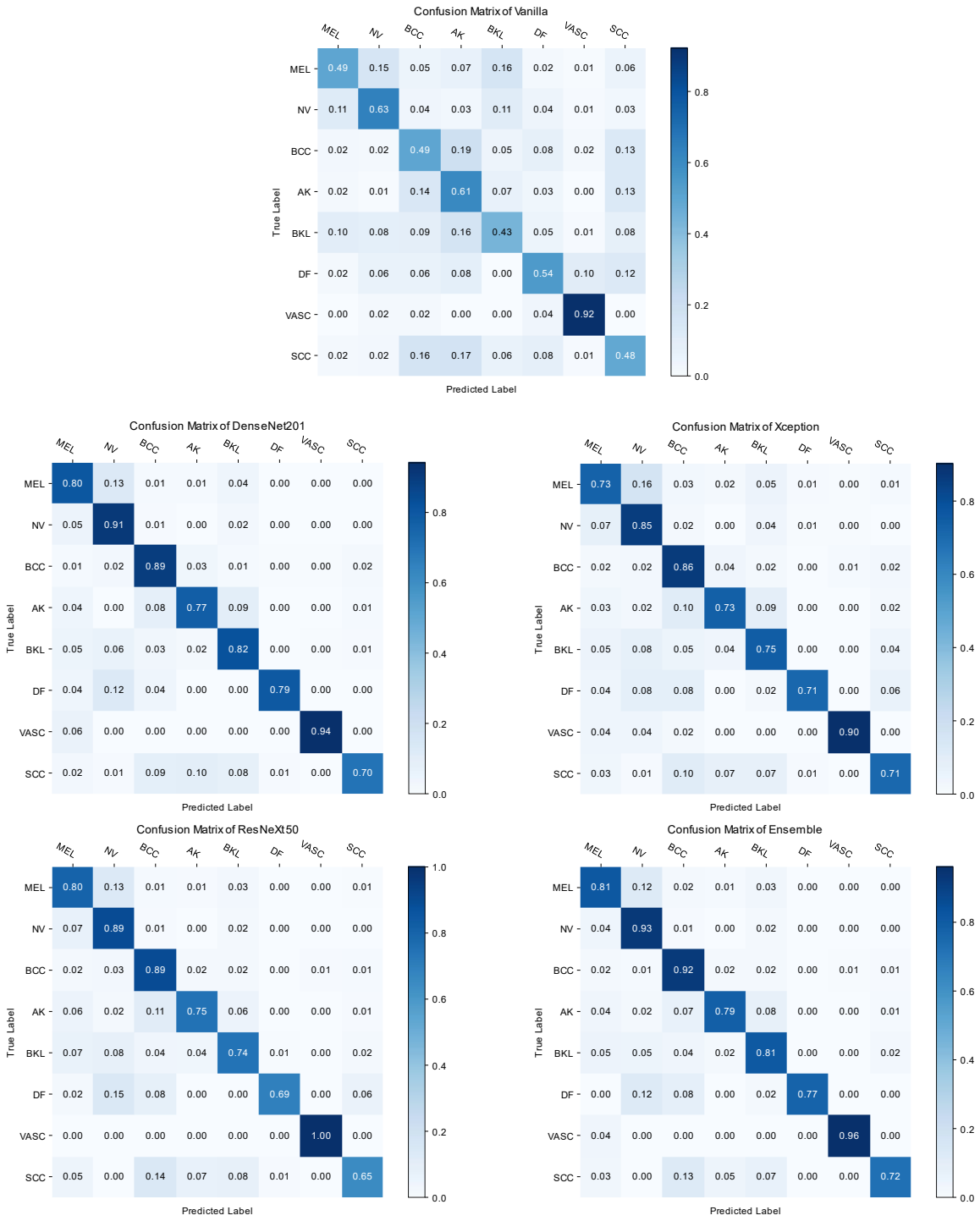


Figure 2 Confusion matrixes of 8 known categories on the validation set

Table 4. Classification performance of 8 known categories on the validation set

Model	Loss	BMCA
Vanilla	1.154	0.576
DenseNet201	0.515	0.828
Xception	0.568	0.780
ResNeXt50	0.536	0.802
Ensemble	0.371	0.838

Comparison between ODIN and Baseline

DenseNet201 model and its softmax scores were used to tune the ODIN parameters since it achieved the highest BMCA as a single model. The optimal parameters that minimize the FPR at TPR 95% are T = 2, ε = 0.0002 and δ = 0.90385. Figure 3 shows the receiver operating characteristic (ROC) curves when using the baseline method [21] and ODIN with optimal parameters. Unlike the results presented in [20], ODIN was only slightly better than the baseline in terms of AUROC. Similarly, as shown in Table 5, FPR was only reduced from 86.57% to 79.1% which is still significantly high.

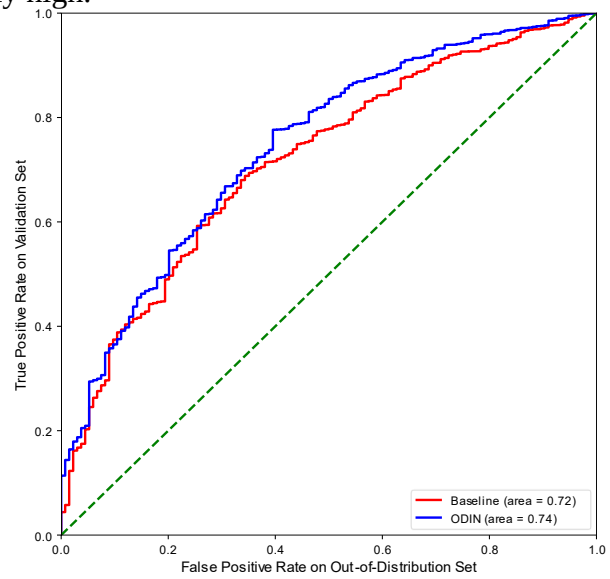


Figure 3 ROC curves of baseline and ODIN

Table 5. Performances of distinguishing in- and out-of-distribution images on the test set

Method	FPR at TPR 95%	AUROC
Baseline	86.57	71.72
ODIN	79.10	74.44

9-category Classification of Approach-1

The ensemble model was integrated with the ODIN method to classify images of the validation set and out-of-distribution dataset together. The BMCA of the ensemble model was dropped from 0.838 to 0.754, and many out-distribution samples were misclassified as other categories especially melanocytic nevus (NV) as shown in the confusion matrix of Figure 4.

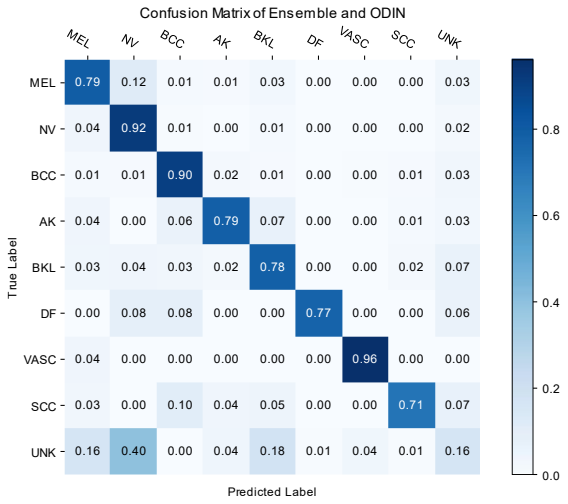


Figure 4 Confusion matrixes of 9-category classification

9-category Classification of Approach-2

The evaluation results of approach-2 on the validation set are shown in Table 6. The ensemble of CNNs slightly outperformed each individual model and achieved a BMCA of 0.804. Figure 5 shows the confusion matrixes of four models over the 9 categories on the validation set. The distributions of misclassification are also similar among models, and the unknown category has the largest portion of images being misclassified. Only about half of the out-distribution samples were classified correctly. Although the ensemble model's BMCA of approach-2 is higher than the corresponding one of approach-1 (0.804 vs. 0.754), it's difficult to judge whether the results will be consistent on the test data.

Table 6. Classification performance of 9 categories on the validation set

Model	Loss	BMCA
DenseNet201	0.523	0.777
Xception	0.583	0.754
ResNeXt50	0.595	0.785
Ensemble	0.377	0.804

Justification

According to the above validation results, it is obvious that all transfer learning models of both approaches outperformed the vanilla model. Furthermore, the simple ensemble method indeed improved the BMCA and achieved a good classification accuracy in terms of known categories. Unfortunately, the ODIN detector performed poorly in distinguishing in- and out-distribution samples unlike those successful cases in previous studies. It might be caused by the fact that the differences between in- and out-distribution samples are insignificant. To verify my implementation of ODIN, I have tried using Iris flower dataset as out-distribution samples and was able to get a very low FPR at TPR 95%. Hence, the implementation of ODIN should be correct. Approach-2 is not a robust solution for identifying the additional outlier class either. As mentioned before, the scope of the outlier is unclear and only 134 images of the out-of-distribution dataset must be far from enough to represent possible categories. A superior novelty detection algorithm is required to tackle this problem.

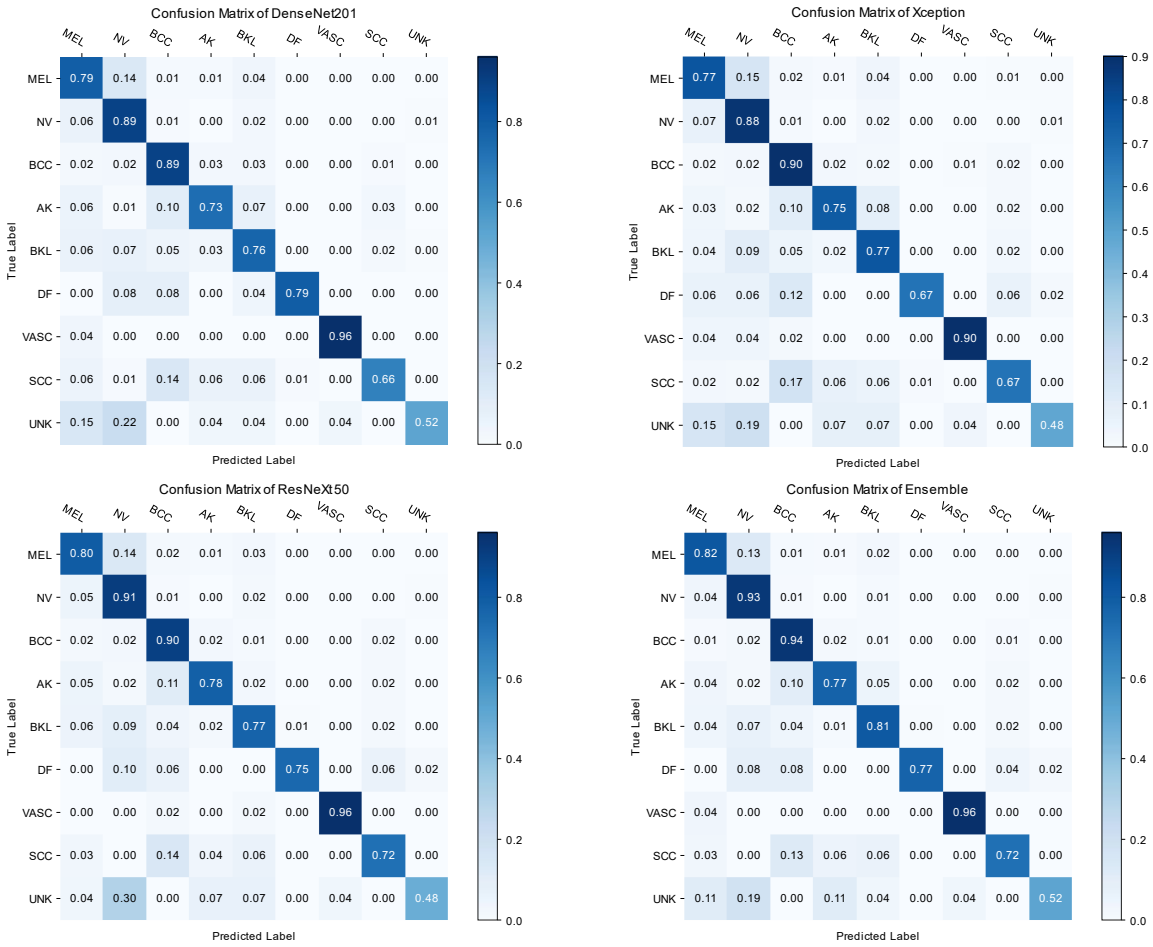


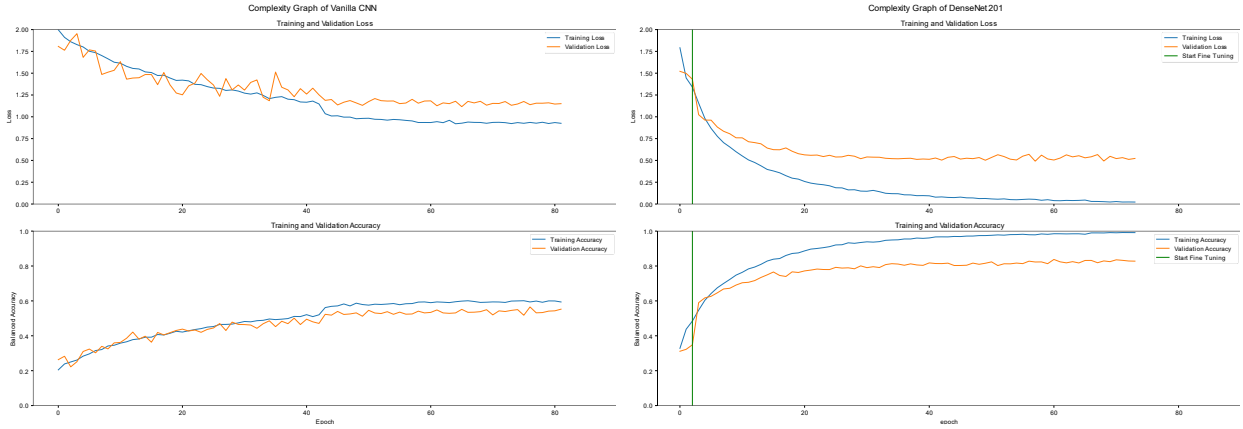
Figure 5 Confusion matrixes of 9 categories on the validation set

Conclusion

Free-Form Visualization

Model Complexity Graph

Figure 6 shows that the vanilla model took more epochs than DenseNet201 to train but converged on much higher loss with lower BMCA. The training loss of DenseNet201 decreased along with the increase of epoch, but the validation loss converged and stopped decreasing from an earlier epoch. I'm not sure if this phenomenon is overfitting.



(a) (b)

Figure 6 Model complexity graph of Vanilla (a) and DenseNet201 (b) regarding loss and BMCA. The blue and orange lines represent training and validation results respectively. The green vertical line denotes the starting epoch for fine-tuning.

Circle Marker

There are almost 200 images having different colors of circle markers in the training data and three of them shown in Figure 7. All these images belong to melanocytic nevus (NV). In approach-1, 30 of these images were split into the validation set, and all of them were correctly classified as NV by all models listed in Table 4 except Vanilla with confidences close to 1 mostly. Even the vanilla CNN only misclassified 2 out of the 30 images. This reminds me of the issue Sebastian Thrun mentioned in lesson 6 that the models might be trained as marker detectors. To check whether the concern really happened, I cropped markers of the three images in Figure 7 out, then classified these cropped images by the DenseNet201 model. Figure 8 shows that the cropped NV images are still classified correctly with high confidences. The concern seems not exist, but of course, more complete analyses need to be conducted to make a solid judgment.

ISIC_0000999_downsampled ISIC_0001385_downsampled ISIC_0002107_downsampled

Figure 7 Dermoscopic images with circle markers.

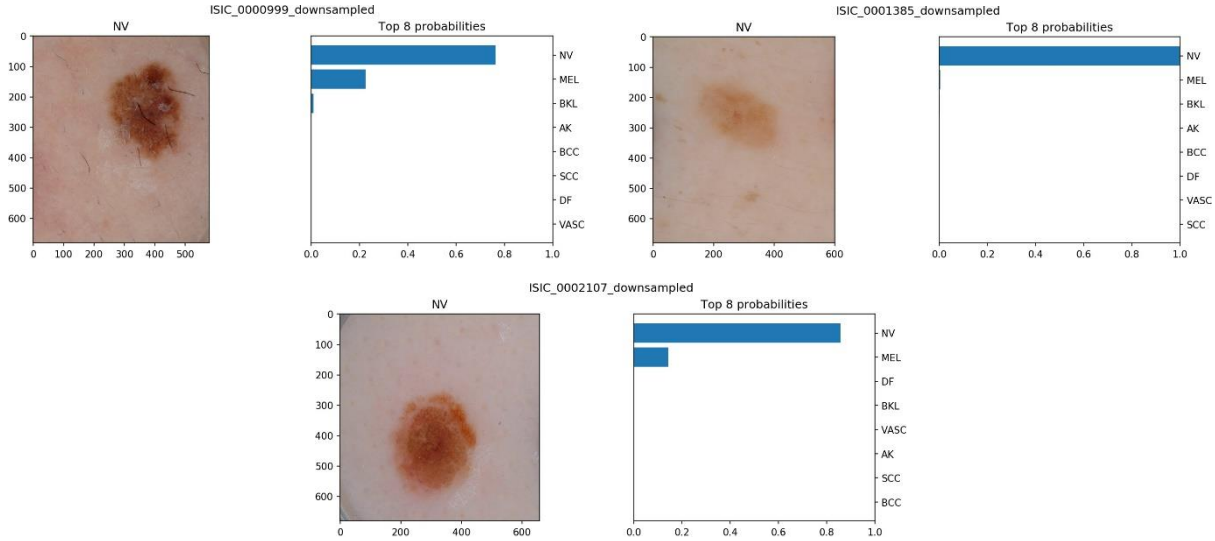


Figure 8 Predicted probabilities of 8 categories for three cropped images.

Reflection

The process used for this project consists of following iterative steps:

1. Understand the ISIC Challenge and relative background
2. Review literature to come up with possible solutions that can be finished within a limited time period
3. Visualize/split the training data and search for other public and free datasets
4. Implement main components and train the vanilla CNN
5. Look for free GPU resources for prototyping and quick implementation verification
6. Implement/train/refine transfer-learning models
7. Implement/train/refine/integrate ODIN method

Implementation

It took much more time than I expected to implement the project. At first, I thought I could just follow examples in Keras document to create and train models. However, there are many functions that I need not provided by the Keras library. It's also not easy to be aware of giving wrong parameter values or incorrect implementation. For example, it took me more than two weeks to realize I should not directly call tf.metrics.mean_per_class_accuracy of TensorFlow in the custom metric function, and all models were required to be re-trained.

Computing Resource

I seriously underestimate the cost of computing and it was one of the biggest obstacles in this project. It made me feel stressful especially when I need to retrain the models. Before training the whole models on expensive cloud services, I moved data and codes around frequently in order to leverage the free GPUs on Google Colab. I have also spent some time comparing different providing solutions of hardware configuration, software environment, region, price and performance among AWS, GCP, and Azure. GCP provides the highest flexibility in terms of hardware configuration that I can customize CPU, memory, and GPU anytime. Besides, the auto-shutdown feature added in *main.py* really helped me save a good fortune. After these, now I understand why the term "AI Arms Race" appears.

Outlier Category

It's surprising that the ODIN detector of approach-1 performed poorly in distinguishing in- and out-distribution samples. Although the results shown in this report are not based on the final test data, I wouldn't expect there will be a satisfactory outcome. I saw someone mentioned about building a binary classifier for each known category with the voting mechanism to solve this kind of problem. Some other novelty detection algorithms might worth trying such as OneClassSVM, One-Class CNN [22] and Open/Pairwise Classification Network [23].

Improvement

Besides novelty detection, there are many improvements that could be made on my approaches. For example, color constancy [24] and lesion segmentation were demonstrated helpful for dermoscopic image classification. Incorporating lesion meta-data could also increase diagnosis performance. Techniques for understanding and visualizing CNNs like layer activations, t-SNE, and saliency maps could help interpret models and identify properties that differentiate among different types of lesions. More lightweight CNN models should be used if the models are going to be deployed on mobile or edge devices.

References

[1] W. C. R. F. International. "Skin cancer." https://www.wcrf.org/dietandcancer/skin-cancer (accessed 08/12, 2019).

[2] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," (in eng), *Nature*, vol. 542, no. 7639, pp. 115-118, 02 2017, doi: 10.1038/nature21056.

[3] Z. Apalla, A. Lallas, E. Sotiriou, E. Lazaridou, and D. Ioannides, "Epidemiological trends in skin cancer," (in eng), *Dermatol Pract Concept*, vol. 7, no. 2, pp. 1-6, Apr 2017, doi: 10.5826/dpc.0702a01.

[4] A. C. Society. "Cancer Facts and Figures 2019." https://www.cancer.org/research/cancer-facts-statistics/all-cancer-facts-figures/cancer-facts-figures-2019.html (accessed 08/12, 2019).

[5] I. S. I. Collaboration. "ISIC Archive." https://www.isic-archive.com (accessed 08/12, 2019).

[6] N. C. F. Codella *et al.*, "Skin lesion analysis toward melanoma detection: A challenge at the 2017 International symposium on biomedical imaging (ISBI), hosted by the international skin imaging collaboration (ISIC)," in *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 4-7 April 2018 2018, pp. 168-172, doi: 10.1109/ISBI.2018.8363547.

[7] P. Tschandl, C. Rosendahl, and H. Kittler, "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions," *Scientific Data*, Data Descriptor vol. 5, p. 180161, 08/14/online 2018, doi: 10.1038/sdata.2018.161.

[8] M. Combalia *et al.*, "BCN20000: Dermoscopic Lesions in the Wild," *arXiv e-prints*,

[9] I. S. I. Collaboration. "ISIC 2019." https://challenge2019.isic-archive.com/ (accessed 08/12, 2019).

[10] I. S. I. Collaboration. "ISIC 2018." https://challenge2018.isic-archive.com/ (accessed 08/12, 2019).

[11] N. Codella *et al.*, "Skin Lesion Analysis Toward Melanoma Detection 2018: A Challenge Hosted by the International Skin Imaging Collaboration (ISIC)," *arXiv e-prints*,

[12] J. Kawahara, S. Daneshvar, G. Argenziano, and G. Hamarneh, "7-Point Checklist and Skin Lesion Classification using Multi-Task Multi-Modal Neural Nets," (in eng), *IEEE J Biomed Health Inform*, Apr 2018, doi: 10.1109/JBHI.2018.2824327.

[13] T. J. Brinker *et al.*, "Skin Cancer Classification Using Convolutional Neural Networks: Systematic Review," (in eng), *J Med Internet Res*, vol. 20, no. 10, p. e11936, 10 2018, doi: 10.2196/11936.

[14] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016, p. 800.

[15] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," (in eng), *Nature*, vol. 521, no. 7553, pp. 436-44, May 2015, doi: 10.1038/nature14539.

[16] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," *arXiv e-prints*,
[17] O. Russakovsky *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *arXiv e-prints*,
[18] P. Tschandl *et al.*, "Comparison of the accuracy of human readers versus machine-learning algorithms for pigmented skin lesion classification: an open, web-based, international, diagnostic study," (in eng), *Lancet Oncol*, Jun 2019, doi: 10.1016/S1470-2045(19)30333-X.
[19] N. Gessert *et al.*, "Skin Lesion Diagnosis using Ensembles, Unscaled Multi-Crop Evaluation and Loss Weighting," *ArXiv*, vol. abs/1808.01694, 2018.
[20] S. Liang, Y. Li, and R. Srikant, "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks," in *ICLR*, 2018.
[21] D. Hendrycks and K. Gimpel, "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks," *arXiv e-prints*,
[22] P. Oza and V. M. Patel, "One-Class Convolutional Neural Network," *IEEE Signal Processing Letters*, vol. 26, pp. 277-281, February 01, 2019 2019.
[23] L. Shu, H. Xu, and B. Liu, "Unseen Class Discovery in Open-world Classification," *arXiv e-prints*,
[24] C. Barata, M. E. Celebi, and J. S. Marques, "Improving dermoscopy image classification using color constancy," (in eng), *IEEE J Biomed Health Inform*, vol. 19, no. 3, pp. 1146-52, May 2015, doi: 10.1109/JBHI.2014.2336473.