American University of Armenia

College of Science and Engineering

Capstone

21.01.2020

Human Skin Cancer Classification Application

Ensemble of Multiple Deep Transfer Learning Convolutional and Residual Neural Networks

Davit Sargsyan
davit_sargsyan@edu.aua.am
Supervisor: Aram Keryan

Abstract

Skin is the biggest human organ. Skin cancer is usually developed from skin color pigments called melanocytes which are present in a regular human skin mole. Thus, in its early stages skin cancer is very similar to an ordinary skin mole, and very hard to differentiate in the early stages of its development. In 2015 there were 3.1 million people with the disease worldwide. Resulting in 59,800 deaths that year [1]. The primary reason for the high death rate is the late detection of the cancer [2]. The aim of this project is to build an ensemble learning based model trained on dermatoscopically analysed and labeled dataset that will be used in a mobile application for detection of skin cancer in a home environment without a need for a doctor.

Keywords: ***Deep Learning, Skin Cancer Classification, Melanoma Detection, Skin Lesion Classification, Image Classification***

Contents

Introduction

Doctors have accuracy* of predicting a skin cancer type with 89% accuracy at most [3] [4], with looking at the mole and taking into account the age, sex of a patient and the position of a lesion. Therefore 11 % of cases are still being mistreated and diagnosed wrongly. The ensemble of models, which will be created as an outcome of this project will provide a better accuracy, by using just the images of the moles. The input of the algorithm is an image of a mole which undergoes several augmentations. Then by using several state-of-the-art models the algorithm returns probabilities of each class of skin cancer. Afterwards the label of the highest probability is given as an output as a diagnosis of a skin cancer. The models are going to be taken with their pretrained weights and fine tuned using images of skin cancer based on a dataset that is verified with dermatoscopic diagnosis. Finally the obtained ensemble of the models will be used in a mobile application which could provide a skin cancer diagnosis instantly by using a camera. Compared to the current dermatoscopic analyses, the newly proposed method will provide an instant result.

*** See Appendix for definition**

Medical Information

Skin is the biggest human organ. Skin cancer is usually developed from skin color pigments called melanocytes which are present in a regular human skin mole. Thus, in its early stages skin cancer is very similar to an ordinary skin mole, and very hard to differentiate in the early stages of its development. In 2015 there were 3.1 million people with the disease worldwide . In the same year, the disease resulted in 59,800 deaths worldwide [1][2].

Types

There are several types of skin cancer. The most widespread types are:

1. Melanoma
2. Melanocytic nevus
3. Basal cell carcinoma
4. Actinic keratosis
5. Benign keratosis (solar lentigo / seborrheic keratosis / lichen planus-like keratosis)
6. Dermatofibroma
7. Squamous cell carcinoma

In early stages these are sometimes indistinguishable for non professional naked eye.

Features

Besides of the dermatoscopic analyses, skin cancer can be detected through the following criteria [5]:

1. *Asymmetry* – The lesion is irregular, or not symmetrical, in shape.
2. *Border* – The edges are irregular and difficult to define.
3. *Color* – More than one color, or uneven distribution of color, exists.
4. *Diameter* – Diameter is greater than 6 mm.
5. *Evolving* – The lesion has changed in color and size over time.

Therefore pushing the model towards detection of these threats can be beneficial. However to be fully sure that the model is detecting the following steps could take years of research. In future works a further study could be done to be sure that those features are detected. Currently it is optimal to leave the Deep Learning CNN do its work and find the features automatically. Indeed, that is the beauty of Deep Learning, that it detects the features it needs for distinction of classes automatically.

Related Work

Skin cancer classification problem is an image classification problem, the main difference is that instead of differentiating between various objects such as a cup or a chair, the problem here is to classify different types of moles which are very similar to each other. There are two main approaches for solving an image classification task: designing an ensemble of neural networks [6][7], or fine tuning a state-of-the-art model [8].

Additionally, The International Skin Imaging Collaboration (ISIC) organizes the competition of skin lesion classification[9]. Hundreds of teams of data scientists from all over the world participate in this competition and share their results. Throughout the research, the high ranking solutions of each year were analyzed in order to understand the approach to the problem and compare it to regular image classification approaches. In almost all cases, the ensemble of CNNs or a single state-of-the-art-model trained with transfer learning was used for approaching the solution of the skin cancer classification problem as it is done in regular image classification tasks.

There are hundreds of works on the topic of skin cancer classification and image classification in general. Some try to solve a binary classification problem to detect whether the mole is malignant or benign. Others solve a ternary classification problem or even a 700 skin lesion classification problem. Below are presented several approaches to the problem grouped by similar methodologies.

Approach 1:

The first approach that was used in many of the researched projects was a use of a single state-of-the-art model alongside with different data augmentations. The important part in each project was the use of a good model. Usually a model such as VGG-16, DenseNet or SEnet154 was being used to solve the skin cancer classification problem. Several models are being trained and the best one is being taken as the final result. Data augmentations are usually related to the resizing of the image to match the input of the corresponding model, as well as some additional noise and rotations being added to the images for regularization. Below are several papers or reports from the ISIC challenge that are using such an approach.

In a paper written by Li Chen [10], breast cancer image classification is conducted using SE-ResNet. Both binary and multi class classification is done. The achieved results for the accuracy of multi class breast cancer image classification were varying in between 90.66% and 93.81%.

Another paper that was investigated was using Google Inception V3 CNN [11]. Random 0-359 degree rotations were applied as a data transformation and the images were resized to be 299x299 to match Google Nets original dimensions. The accuracy of the model on the dataset was around 72.1% in their tests. However instead of classifying only skin cancer types, they were classifying more than 700 different skin lesion types, where some of the lesion types were skin cancers as well.

In a report done by Mengting Xu, Tao Zhang, and Zhongnian Li [12] VGG-16 and CAVGG-16 were being trained and the best result (VGG) was being taken. In another report by

Anand Nambisan [13], ResNet was used for training on the dataset, where images are resized to match the input of the model, as well as sharpening, blurring and flipping are applied to the images to avoid overfitting.

In ISIC Challenge [14] dozens of similar approaches are described, where a state-of-the-art model is fine tuned on the augmented skin cancer dataset.

Approach 2:

The second approach that was found was a logical continuation of the first approach. As in the previous approach, here again several state-of-the-art models were fine-tuned, however instead of taking the best model as a final result, top performing 3-5 models were being chosen and the ensemble of those models was being constructed in order to boost the overall performance on the skin cancer classification or general image classification problem. Below are presented several similar papers which were solving the problem with using such an approach.

F. Codella[15] describes a usage of deep learning ensembles for melanoma recognition. On the same dataset 6% higher accuracy is being achieved with their method compared to the result of expert doctors. The achieved accuracy of 76% was achieved on a dataset containing 900 training and 379 testing images.

In his paper “An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification”[7] Ashnil Kumar describes a similar approach of using an ensemble of CNN’s to conduct a medical image classification in general. It is hypothesized there that as different CNN’s learn different levels of semantic image representation, the ensemble of such models could achieve a better result on medical data.

The approach which gave the best metric value in ISIC Challenge was using an Ensemble of Multi-Res EfficientNets and SEN154 2 [16]. The Primary metric that was used there was the balanced multiclass accuracy**. This method gave a 0.636 Balanced Multi Class Accuracy. Another result was achieved by using the Ensemble of Efficient B3, B4 and Seresnext101 [17]. Similar result was obtained with these models as well: 0.607 Balanced Multi Class Accuracy.

A team from Autonomous University of Barcelona has reached a validation score of 0.76 in skin cancer classification problem [18]. They have tried different single state-of-the-art classification models in their solution and also the ensemble of those models. They used different data augmentation techniques before feeding the data into models, specifically: rotation, flip, random crop, adjust_brightness, adjust contrast, pixel jitter, Aspect Ratio, random shear, zoom, and vertical and horizontal shift and flip. PNASNet-5-Large, InceptionResNetV2, SENet154, InceptionV4 and the Ensemble of those models were tested and the best result was achieved with the SENet154 model. The training technique that they have used in their work is freezing all the layers of models, training the last layer and then unfreezing all the layers and fine-tuning.

Strengths / Weaknesses

Although in the discussed papers there were two major approaches, data augmentation was done similarly in most of the cases. Random rotations, image resizing for matching the input of the used model, cropping, flipping, noise addition, blurring and sharpening was being applied to each image. This was a strength in all of the papers as it helps to overcome the problem of

**** See Appendix for the definition**

overfitting the dataset, therefore improving the accuracy on the testing set.

Use of a single model was solving the classification problem, however the achieved accuracy with this approach was not as high as the ensemble approach. A single model can classify accurately most of the classes and miss some of them. If using several models with ensemble learning this problem will also be solved, as the cases where one model has a low accuracy of prediction will be compensated by the accuracies of the models which predict the same class better. That is the advantage of ensemble learning, as it uses the average of the predictions of each class correspondingly. Therefore it is better to go towards the solution of using an ensemble of several state-of-the-art models.

Dataset

There were several datasets available in various sources. The main problems with most of the datasets was either the absence of multiple classes, i.e. the datasets were intended for a binary classification problem, or the datasets were not verified through dermatoscopy by licensed doctors. Only 2 datasets were shared publicly and were satisfying the criteria mentioned above.

The first option is to take the dataset of The International Skin Imaging Collaboration [19], which contains more than 23000 color images of resolutions varying from 500x500 to 1650x962. The diagnosis for each picture is given by expert doctors after dermatological analyses, therefore all images are almost surely correctly labeled.

Here are the labels and number of pictures in each class:

nevus: 18566; ***melanoma***: 2169; ***pigmented benign keratosis***: 1099; ***basal cell carcinoma*** : 586; ***seborrheic keratosis***: 419; ***squamous cell carcinoma***: 226; ***vascular lesion***: 142, ***actinic keratosis***: 132; ***dermatofibroma***: 122; ***lentigo NOS***: 71; ***solar lentigo***: 57; ***lentigo simplex***: 27; ***angioma***: 15; ***atypical melanocytic proliferation:*** 13; ***other:*** 10; ***angiofibroma or fibrous papule:*** 1; ***scar***: 1; ***lichenoid keratosis***: 1.

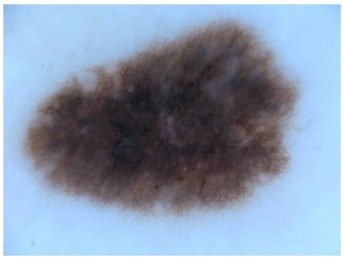


Figure 0. Sample Image from the dataset

The first 9 classes will be taken and the rest of the images will be treated under the "other" class. Classes after the 9th class in ISIC dataset contain 1, 10, 13 or 15 images. Oversampling such classes is not beneficial as it would create biased results. Therefore it is better to unite such classes in a "other" class. It would also help to generalize the "other class" as it would have different types of image classes in it. This and other techniques done to solve the class imbalance problem are described under the sections Data augmentation and Class Imbalance. Nonetheless, ISIC dataset has collaborators from all over the world and in future when there are enough images from other classes, they will also be included in the classification.

The second option is to go with the HAM-10000 dataset [20] composed of RGB images of resolutions varying from 500x500 to 1650x962, which is a subset of the ISIC dataset with fewer classes. Mostly the lightest and clearest images are included in that dataset. The types of lesion and the number of its representative images are the following: Dermatofibroma (**df**) - 115 images, Vascular skin lesion (**vasc**)-142 images, Actinic Keratoses and intraepithelial Carcinoma (**akiec**) - 327 images, Basal cell carcinoma (**bcc**)-514 images, Benign keratosis(**bkl**)-1099 images, Melanoma (**mel**)-1113 images and Melanocytic nevi (**nv**) - 6705 images.

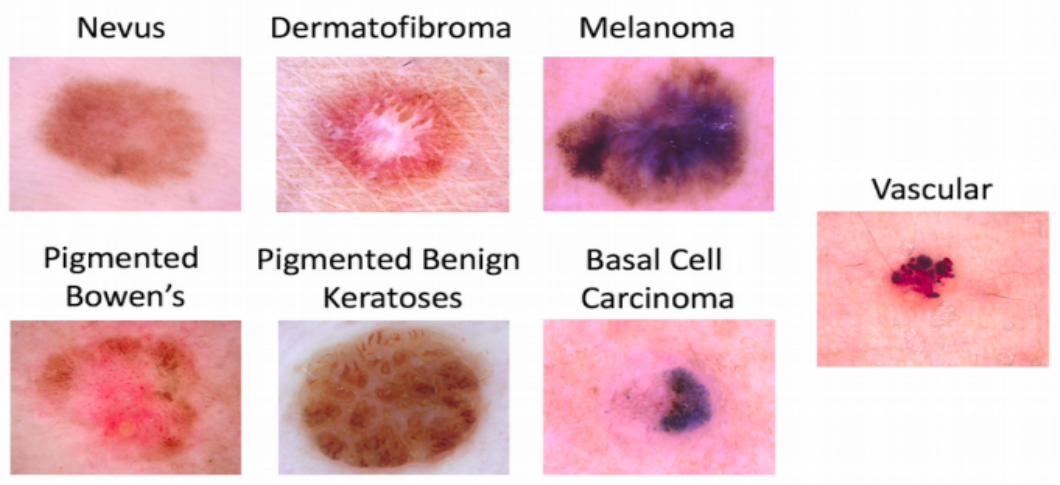


Figure 1. Samples from each class from the HAM10000 Dataset.

During all of the training data 20% of data is left for testing. The rest of the data is splitted as 80% training and 20% validation sets. Although both of the datasets were being used on the initial training, the final training is going to be done on the HAM 10000 dataset. Images in the HAM10000 dataset have better quality than the ones from the ISIC Dataset. ISIC Dataset has more minor classes which makes the class imbalance problem even more tangible there. In future when the ISIC dataset would accumulate more images for all of the classes, then the training would also be done on that dataset during the future works.

Currently it is not planned to use other features such as age or gender of a patient and will rely only on the features that are automatically extracted from the images through our deep learning CNNs.

Data augmentation

Since the datasets have a "class imbalance problem" (some of the classes are several times bigger than the others, e.g. nv has 6705 images and df has 115 images), data augmentation

is essential to apply in order to increase the number of images in the minor classes (classes which have lesser images). Various image augmentations were applied. According to the study of Mateusz Buda, Atsuto Maki and Maciej A. Mazurowski **[21]** there are two common techniques for handling the class imbalance problem- *undersampling and oversampling*. Oversampling will be taken as a solution for two major reasons. Firstly with undersampling a certain amount of information is being lost. Secondly a study about class imbalance **[21]** shows that in the majority of cases oversampling works better than other methods.

So as a solution, random images from minority classes were taken and copied with the use of augmentation techniques.

Below you can find the list of data augmentation techniques that were applied for oversampling:

***Vertical flipping, random zoom, random crop, random rotations*.**

After the oversampling of HAM10000 dataset the number of images in each class became balanced:

df - 5058 images,

vasc - 5021 images,

akiec - 5112 images,

bcc - 5114 images,

bkl-5195 images,

mel - 5287 images,

nv - 5364 images.

And similarly ISIC Dataset has changed and each class became balanced:

nevus: 18566;

melanoma: 18554;

pigmented benign keratosis; 18550;

basal cell carcinoma : 18547;

seborrheic keratosis: 18516;

squamous cell carcinoma: 18537;

vascular lesion: 18556,

actinic keratosis: 18534;

dermatofibroma: 18543;

other: 18501

Additionally the whole dataset underwent through the following augmentations:

adjust_brightness, adjust contrast, pixel jitter, perspective warp .

Problems

1. Downloading the dataset.

There was a download functionality on the ISIC Archive website which was not working. There was an open source code in github [39] which failed every time in around 50% of execution.

So I have decided to write my own downloader [22] using the provided API [23].

2. Image sizes

Although data is from one source, it varies in resolution and dimensions. E.g. there are images of such shapes:

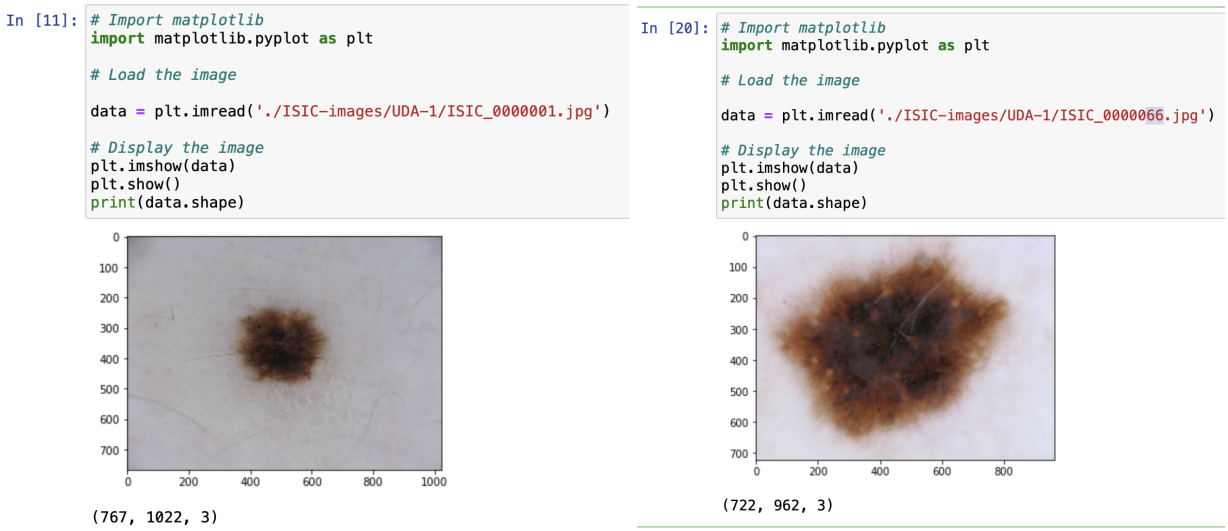


Figure 2. Figure 3. Sample Images from the dataset

The problem is solved during data augmentation where images are resized to the same size based on the model they are going to be feeded in.

3. New image quality

Images taken by regular people for scan may have many problems, such as:

1. Low resolution
2. Low lighting
3. High contrast
4. Blurriness
5. Shades
6. Extra Elements
7. Etc.

The problem is planned to solve out of the scope of this project. If such problems are detected in an image scanned with the application, then the application should require additional scanning for a better quality image.

4. Skin types

Skins of people who belong to different race groups are different. Because of that the model which is trained using pictures of Europoid people's pictures will work with high accuracy for people of European descent and will predict less accurately for people from other races such as Chinese or African People.

The problem is planned to solve out of the scope of this project. The possible methodologies will be:

- to have a branched model for different skin types
- To train the same model with different skin type images and use corresponding model for corresponding skin color/type
- Generalize the model in a way that it would distinguish between skin colors/types by using different skin type images in the dataset.

5. Misleading Features

Each image from the dataset has characteristics other than the cancer itself on it. The images shown below in Figures 4 and 5 have hair and some bare skin parts on them. Other images may also include veins or some unnecessary data on them. If this kind of information is present on most of the images of the dataset, then there is a risk that the model will also learn those features as well.

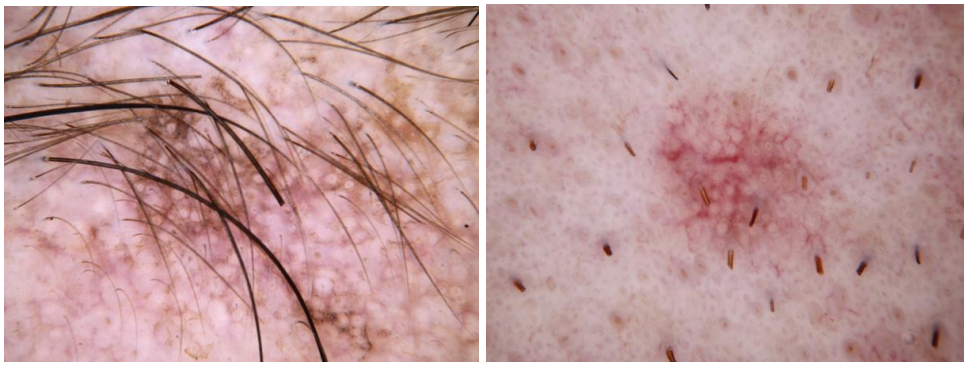


Figure 4. Figure 5. Sample Images from the dataset.

A proposed solution to this problem could be through data augmentation, where through another deep learning model, the cancerous part of the skin will be detected and cropped circularly in order to include only the necessary information in the image as it is shown in the Figure 6 below.

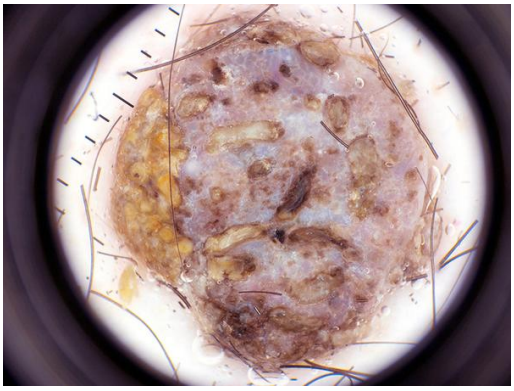


Figure 6. Cropping of an image by Detection of a cancerous region and feeding of the cancerous part to the classification model.

However this is a whole new topic for research and is planned to investigate out of the scope of this project.

6. Overfitting

The base model that was created for initial training was overfitting the data pretty easily. Standart Deep Learning anti-overfitting techniques such as ***data augmentation and regularization (Fastai L2 regularization)*** were applied to get rid of the overfitting and have a good result on the validation set.

7. Model Capacity

The initially created model was limited in its capacity and even when increasing the amount of convolutional and fully connected linear layers, the result was not improved substantially. 78% of accuracy was being achieved on the validation set, which is less than the 89% prediction accuracy of dermatologists [3][4]. This was one of the reasons to research and

find further studies related to the same problem described in the section Related Work and create a solution which is inspired by the related works.

8. Class Imbalance

Perhaps the most important problem of the project was the class imbalance. As seen below in Figure 7, the first 3 classes in HAM 10000 dataset (nevus,melanoma and bkl) comprise 89% of the dataset.

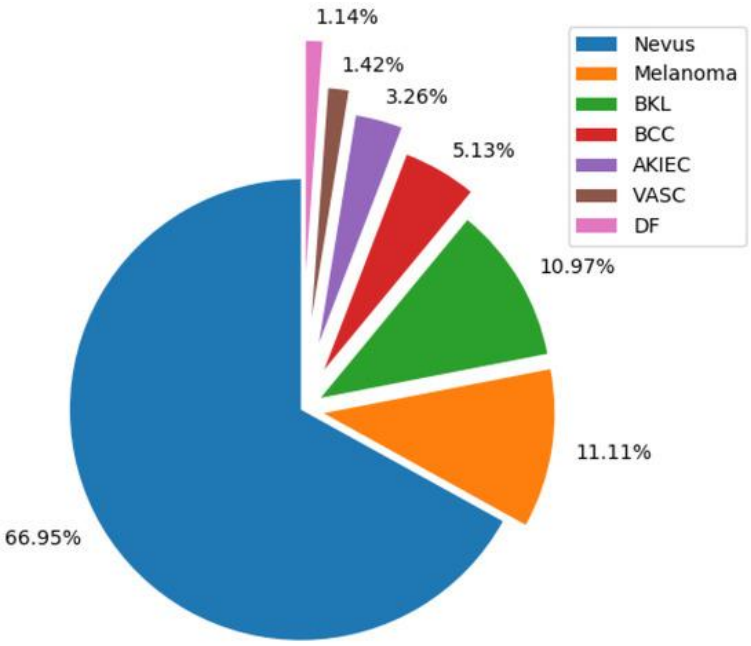


Figure 7. Class Imbalance Pie Chart

It was very easy to obtain high ***accuracies*** on both datasets. However ***F 1 score**** *and* ***balanced multi class accuracy*** [24][25][26] were not high.

*** See Appendix for the definition**

This was a clue that some of the classes which contain few pictures were predicted to belong to a wrong class. There were several solutions to the problem, which are listed below.

1. Search for a bigger dataset with balanced classes.
2. SMOTE Methodology [27].
3. Oversampling [43]. More on oversampling is described under Data augmentation section.
4. Undersampling [44].
5. Uniting the classes with the “other” class if the number of images in the class are less than a fixed amount of some X amount of images. In the case of the ISIC Archive dataset this number could be 100, as the classes in the dataset with less than 100 elements have only 40, 20 or even 1 element. Oversampling of such classes is not beneficial as oversampling a minority class which has a single image would not be beneficial and may create biased results.

The chosen solutions and the decision behind the choice are described under the section Data Augmentation.

Methods

In the beginning it was decided to create a convolutional neural network which would solve the whole classification task, however relying on the related works on the same problem, it was decided to take the path of selecting the state-of-the-art image classification models and training them on the HAM10000 dataset using Transfer Learning. Some of the solutions of the same problem were done by training big state-of-the-art models on the dataset such as Efficient Net B7, however because of the hardware limitations it was not possible to train such models. Training was done using NVIDIA GTX 1060 GPU, RTX 2080 and Google Colab servers. Because of such limitations it was decided to use smaller versions of the models that are described in related works. I have trained several image classification pretrained models with the following technique:

1. The last layers of the pretrained models were adjusted under the number of classes that our datasets have.
2. Then, during the first (10) epochs training was done with frozen layers except for the last one.
3. Afterwards, all layers were unfrozen and the training continued through fine-tuning.
4. The last step was to select those models that have achieved better accuracy, than the baseline accuracy of 0.89 (this was the maximal accuracy that doctors achieved), and generate the final output by their ensemble model.

The best performing models were EfficientNet-b0, EfficientNet-b1, ResNeXt26 and ResNeXt50. EfficientNet models take 448x448 sized images as input and ResNeXts take 299x299 sized input.

Ensemble Learning

Different CNN's learn different levels of semantic image representation [7]. Each model architecture is better in recognizing some specific types of patterns better than other patterns. Using an ensemble helps to take an advantage out of this. Ensemble model uses the results of multiple classification models to obtain a better prediction that each of the used models could have obtained if trained and used distinctly **[28][29][30].** Instead of taking an output of predictions of each class from a single model, ensemble learning takes the averages of predictions of each class and achieves a better result in most of the cases (See Figure 8). Such a result is achieved because in cases where one model shows bad results on predicting a specific class, that is being compensated by another model which has better prediction results for that specific class.

In this project the ensemble of several state-of-the-art models was used to obtain better results than was obtained in each model. The outputs of each model, (the vector with 7 elements representing the probabilities for each of the 7 classes) were summed up and divided by the number of models (4). Finally, from the resulting vector the element with maximum value is chosen and its corresponding class is served as a prediction of the ensemble model. A visual representation can be seen above:

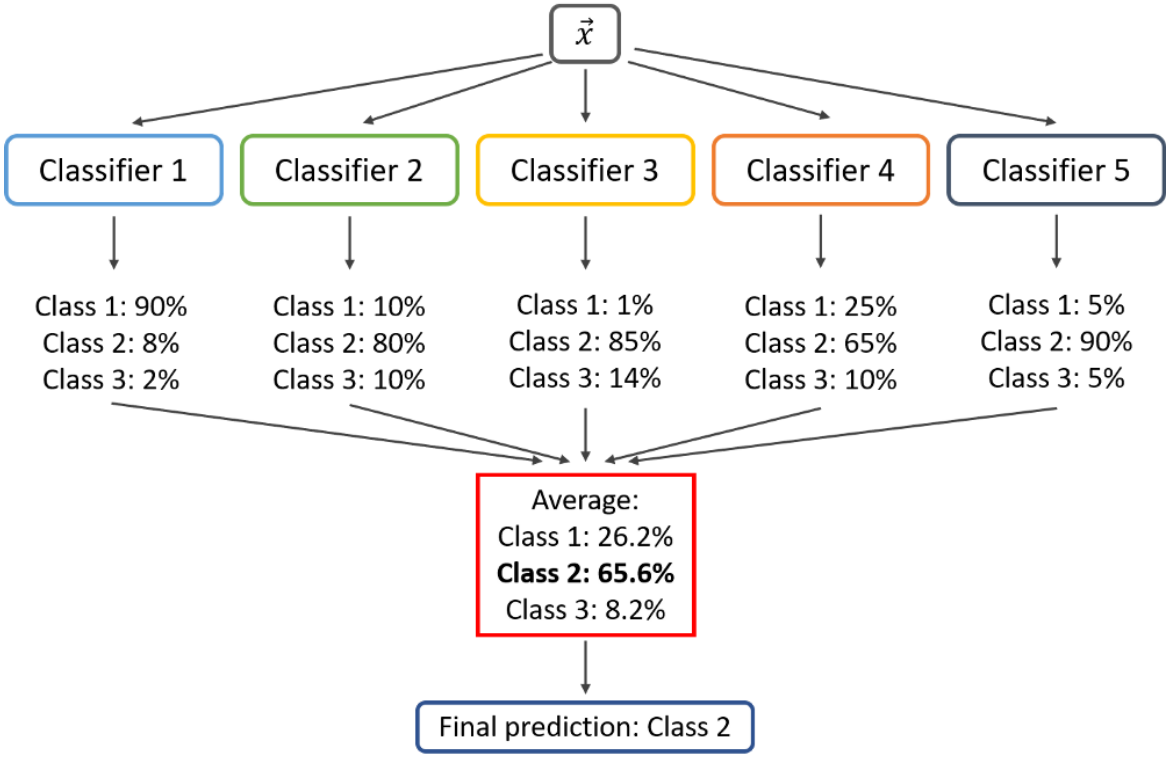


Figure 8. Ensemble Learning Visual Representation of models and averaging Ensemble Learning Techniques— VotingClassifier – mc.ai

The loss function was chosen to be the Cross-Entropy and the optimizer was chosen to be the ADAM.

$$CCE(p,t) = -\sum_{c=1}^{C} t_{o,c} \log\left(p_{o,c}\right)$$

Instead of training the model with a traditional strategy of fixing the hyperparameters and training, the method of one cycle policy was used proposed by Leslie N. Smith [30]. After the range of learning rate is selected (the algorithm of selection is described under the section Results) a cyclical training was performed, during which the training runs with different learning

rates starting from the minimum value of the chosen range of learning rates to the maximum value.The main idea is that higher learning rates allow to jump over local minima points and not stuck there, lower learning rates allow not to jump over the minima. In cyclical learning alongside the learning rate, the momentum also keeps changing in a selected range. However, unlike the learning rate, the range of momentum values stays fiex between 0.85 and 0.95, as suggested in Smith's paper [30]. The value of momentum keeps increasing when the learning rate decreases and vice versa. The intuition behind this technique is that in those parts of training when the gradient tends to find the good area faster, the weights of it are increased by increasing the value of momentum.

Experiments, Results, Discussion

During the stage of choosing a model, various models were trained on the dataset based on the research of related works: EfficientNet-b0, EfficientNet-b1, ResNeXt26 and ResNeXt50, Resnet34, Resnet50, VGG16, VGG19, Google Net, Inception Net V3. The choice of the pretrained models was based on three major factors. The first was its common appearance in other solutions of this problem. The second one was the good performance of the model on the famous image classification challenge on ImageNet dataset. And the third factor was the size of the model which can be handled by the available hardware.

Models

ArmNet

The Model shown below was created as a trial with few convolutional layers. It was created from scratch to understand what kind of accuracy can be achieved through few convolutional layers. The model was overfitting on the HAM 10000 Dataset in the beginning. Training was done on the HAM10000 Dataset with basic augmentations: ***resize, center_crop and random_rotations(20degree)***. Several Trials were conducted:

Trial 1:

```
class ArmNet(nn.Module):
 def __init__(self):
   super().__init__()
   self.conv1 = nn.Conv2d(3, 30, 5, 1)
   self.conv2 = nn.Conv2d(30, 100, 5, 1)
```

```
  # self.bn1 = torch.nn.BatchNorm2d(30)
  self.fc1 = nn.Linear(53*53*100, 800)
  self.dropout1 = nn.Dropout(0.7)
  self.fc2 = nn.Linear(800, 550)
  self.fc3 = nn.Linear(550, 8)
  # softmax
def forward(self, x):
    x = F.relu(self.conv1(x))
    # x= self.bn1(x)
    x = F.avg_pool2d(x, 2, 2)
    x = F.relu(self.conv2(x))
    x = F.max_pool2d(x, 2, 2)
    x = x.view(-1, 53*53*100)
    x = F.relu(self.fc1(x))
    # x = self.dropout1(x)
    x = self.fc2(x)
    x = self.dropout1(x)
    x = self.fc3(x)
    return x
```

Figure 9. Figure 10. Loss/Accuracy plots

Trial 2:

This time the training was done on the ISIC Archive dataset. Information about the dataset and what data augmentations were applied can be found under the section Dataset.

```
class ArmNet(nn.Module):
def __init__(self):
  super().__init__()
  self.conv1 = nn.Conv2d(3, 30, 5, 1)
  self.conv2 = nn.Conv2d(30, 100, 5, 1)
  # self.bn1 = torch.nn.BatchNorm2d(30)
  self.fc1 = nn.Linear(53*53*100, 800)
  self.dropout1 = nn.Dropout(0.7)
  self.fc2 = nn.Linear(800, 550)
  self.fc3 = nn.Linear(550, 10)
def forward(self, x):
    x = F.relu(self.conv1(x))
    # x= self.bn1(x)
    x = F.avg_pool2d(x, 2, 2)
    x = F.relu(self.conv2(x))
    x = F.max_pool2d(x, 2, 2)
    x = x.view(-1, 53*53*100)
    x = F.relu(self.fc1(x))
    # x = self.dropout1(x)
    x = self.fc2(x)
    x = self.dropout1(x)
    x = self.fc3(x)
    return x
```

```
epoch:  1
training loss: 0.7100, 77.6418
validation loss: 0.6982, 78.1426
epoch:  10
training loss: 0.3841, 86.6091
validation loss: 0.6229, 80.1297
epoch:  20
training loss: 0.0311, 99.2732
validation loss: 1.1150, 79.6486
epoch:  30
training loss: 0.0148, 99.5869
validation loss: 1.3287, 79.6695
```

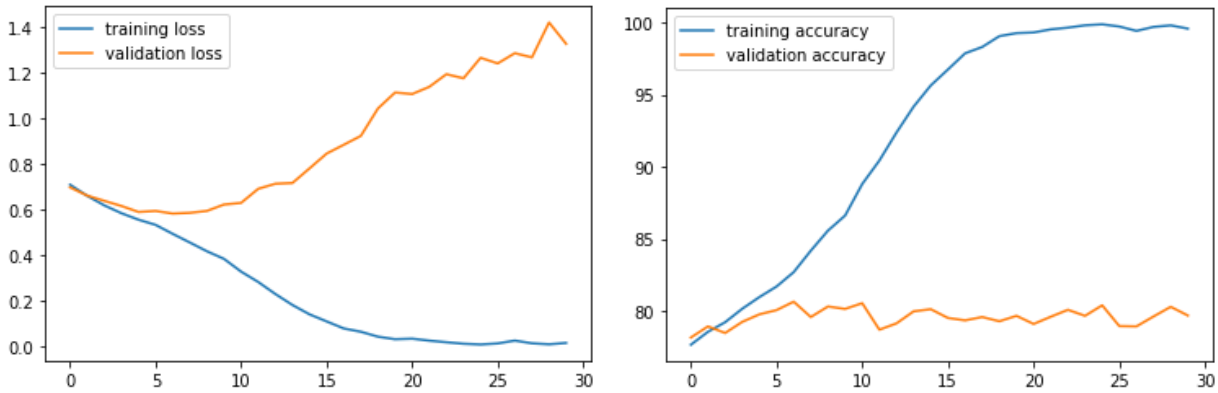


Figure 11. Figure 12. Loss/Accuracy plots

Trial 3:

After applying more regularization techniques by using data augmentation, it was possible to achieve almost the same validation accuracy of 78% as it was obtained during the training of Google Net and Inception Net on HAM10000 and ISIC datasets. The notable fact here is that the same 78% validation accuracy was achieved with a model which is several times smaller than Google Net or Inception Net. However, such a number is even lower than the accuracy achieved by dermatologists, which was another reason to continue the research towards the related works and use an approach that should hypothetically give a higher result.

Changes:

1. Random Rotation of +-180 degrees were applied to the data
2. Random translation of 0.15 was applied
3. Softmax layer was added to have a probability vector output in a range of 0-1, and sum of elements of 1. For classification problems softmax guarantees to return the probability of the highest class.

$$\sigma(x_j) = \frac{e^{x_j}}{\sum_i e^{x_i}}$$

```
class ArmNet(nn.Module):
        def __init__(self):
          super().__init__()
          self.conv1 = nn.Conv2d(3, 30, 5, 1)
          self.conv2 = nn.Conv2d(30, 100, 5, 1)
          # self.bn1 = torch.nn.BatchNorm2d(30)
          self.fc1 = nn.Linear(53*53*100, 800)
          self.dropout1 = nn.Dropout(0.7)
          self.fc2 = nn.Linear(800, 550)
          self.fc3 = nn.Linear(550, 10)
          self.softmax = nn.Softmax(dim=1)
        def forward(self, x):
           x = F.relu(self.conv1(x))
           # x= self.bn1(x)
           x = F.avg_pool2d(x, 2, 2)
           x = F.relu(self.conv2(x))
           x = F.max_pool2d(x, 2, 2)
           x = x.view(-1, 53*53*100)
           x = F.relu(self.fc1(x))
           # x = self.dropout1(x)
           x = self.fc2(x)
           x = self.dropout1(x)
           x = self.fc3(x)
           x = self.softmax(x)
           return x
```

training loss: 1.6852, 77.6157
validation loss: 1.6827, 77.8498

ResNet50

The first model that was used on the dataset is ResNet50. The model was chosen based on its good results on ImageNet dataset and its frequent use in the researched papers. The model showed a 0.831253 accuracy on validation set and had a place for improvement, however the f1 score was low 0.685555. This can be noticed from the confusion matrix shown below, where the dominant class (nv- nevus) is predicted relatively accurately, however classes such as bcc and bkl have many images predicted wrongly.

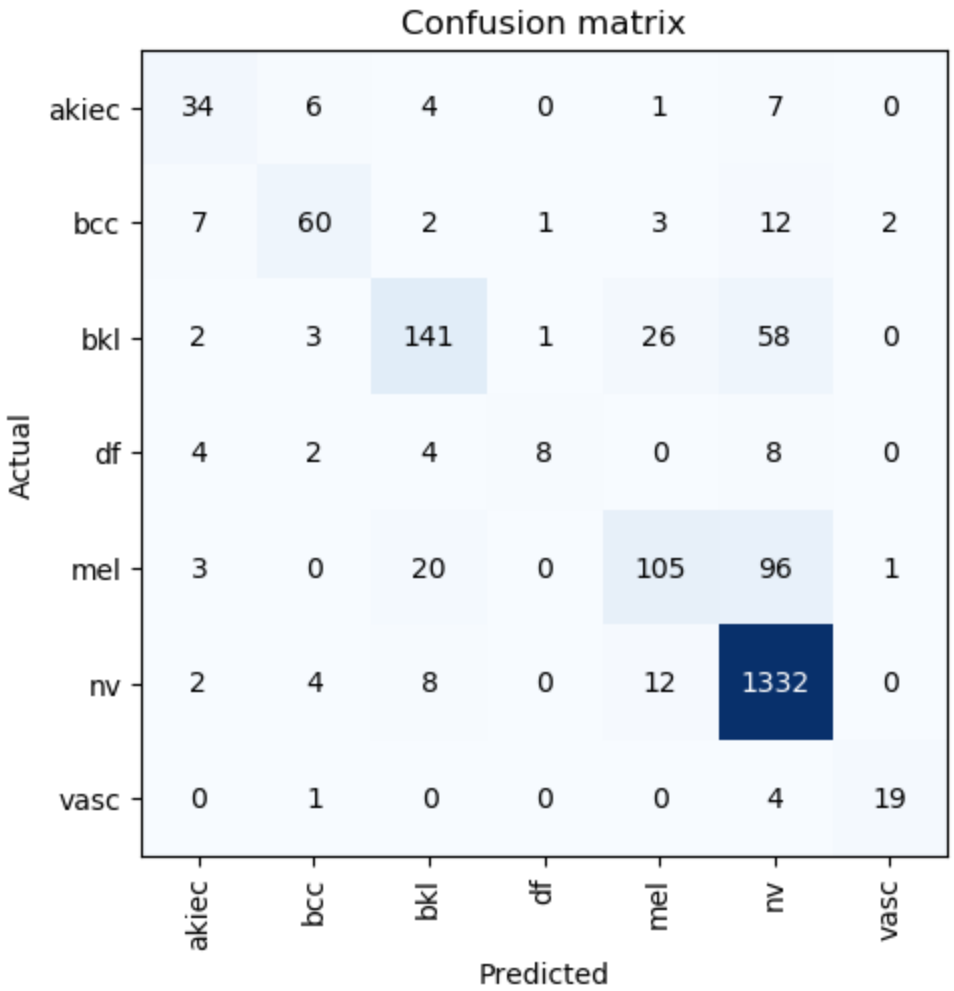


Figure 13. Confusion Matrix of ResNet Training

VGG

VGG-19 is the next model that was used for training. The model was chosen out of curiosity as it has fewer layers than the rest of the models and if it showed good results, then it could be a better use in mobile applications thanks to its smaller size then the rest of the studied models. Similar results to ResNet50 were achieved: 0.81 accuracy and 0.67 f1 score.

Google Net

Google Net was chosen to be tested based on a paper discussed in section Related Work

Trial 1:

epoch: 1

training loss: 1.8431, 75.7176

validation loss: 1.7698, 78.4146

epoch: 2

training loss: 1.7784, 77.4745

validation loss: 1.7685, 78.4146

Trial 2:

It was decided to remove the dropout layer, train several epochs with frozen pretrained model parameters to avoid backpropagation through them and then unfreezed and continue the training. Applying even more rotations to each image (0-359 degree) we see that the model shows better results on the validation set compared to the training set.

Frozen:

epoch: 1

training loss: 1.8332, 77.0771

validation loss: 1.7695, 77.7662

epoch: 2-10

training loss: 1.7731, 77.6366

validation loss: 1.7709, 77.7662

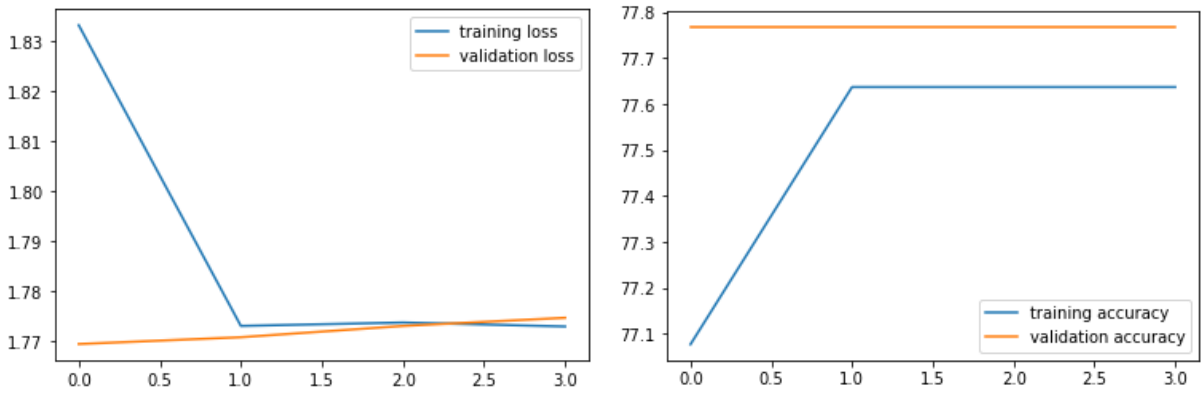


Figure 14. Figure 15. Loss/Accuracy plots

Unfrozen:

epochs: 1-19

training loss: 2.2274, 77.6366

validation loss: 2.2962, 77.7662

Inception V3

Inception Net was chosen to be tested based on a paper discussed in section Research and Trials Batch size was reduced from 100 to 50 to fit into memory when training with unfrozen Gradients. The results of the model were almost identical to the Google Net, described above.

Efficient Net

Efficient Nets were chosen based on their astonishing results on Image Net compared to other state-of-the-art models [31]. Most of the efficient nets provide a better accuracy in the same time being smaller than the alternatives. As the model is going to be used inside of an application, the size of it is very important.

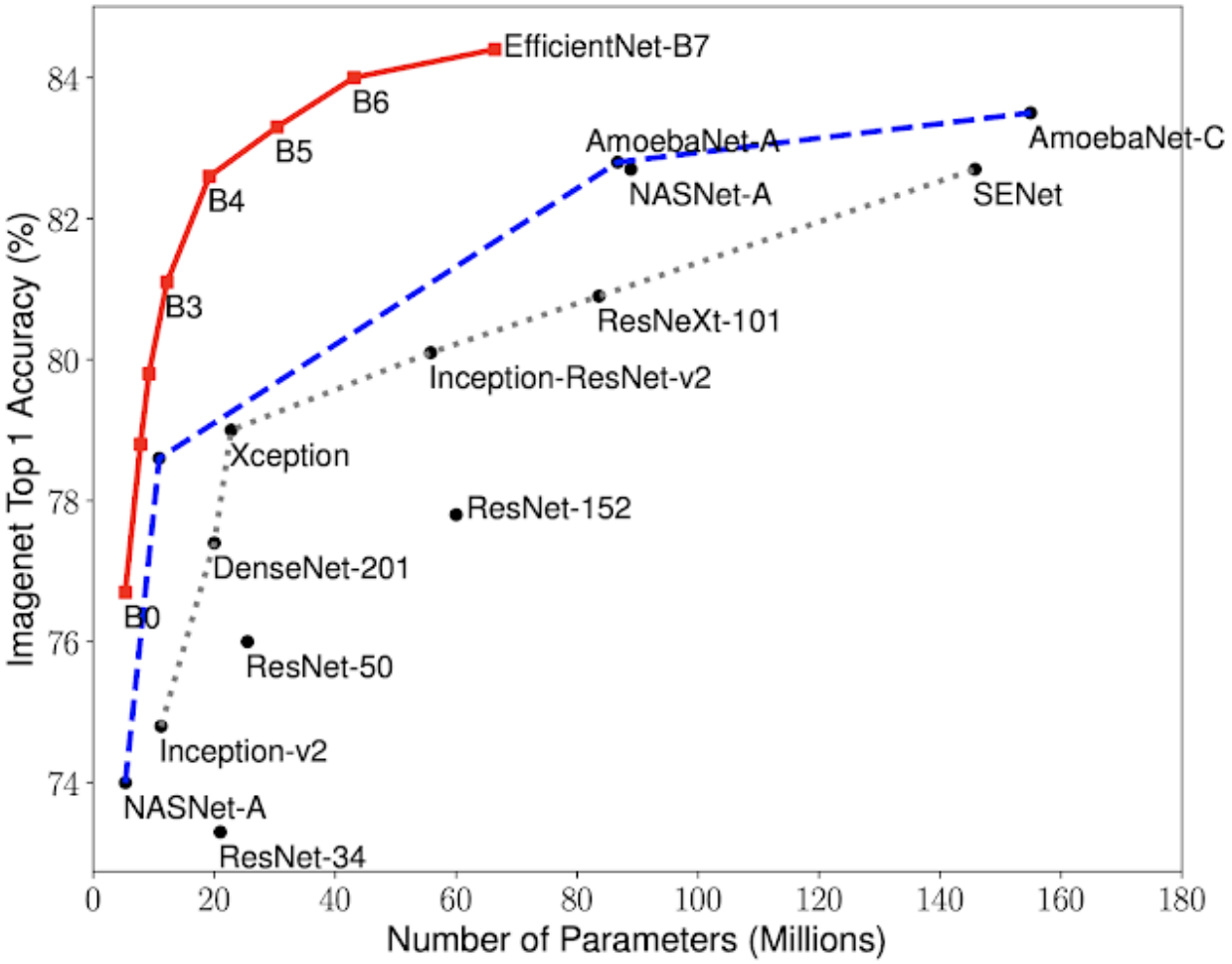


Figure 16. Comparison of the efficient nets to other state-of-the-art models. Size-Accuracy representation [31]

The pretrained models of Efficient Net B0 and Efficient Net B1 were loaded and trained with frozen layers for 10 epochs, after which around 30 epochs were run with unfrozen layers. Batch size was limited to 20 due to limited hardware. If trained with a greated batch size and more epochs, a higher accuracy could have been easily achieved, as even during the last epochs the results were still converging to a lower loss both on training and validation sets. See below

screenshots taken during the training of Efficient Net B0 after changing the learning rate several times:

epoch	train_loss	valid_loss	f_beta	accuracy	time
0	0.797011	0.584507	0.386634	0.806944	07:16
1	0.594157	0.536215	0.365564	0.810082	07:17
2	0.576240	0.474314	0.465782	0.826814	07:28
3	0.493611	0.493933	0.445386	0.829952	07:36
4	0.428174	0.408832	0.529964	0.852750	07:57
5	0.420587	0.410443	0.536170	0.848358	07:55

epoch	train_loss	valid_loss	f_beta	accuracy	time
0	0.295805	0.344238	0.639453	0.869274	07:15
1	0.304530	0.338586	0.640238	0.871366	07:27
2	0.294300	0.337526	0.635358	0.871784	07:40
3	0.301623	0.327662	0.656862	0.878896	07:55

epoch	train_loss	valid_loss	f_beta	accuracy	time
0	0.296112	0.324247	0.647045	0.877850	07:16
1	0.269307	0.322231	0.651089	0.877850	07:15
2	0.265956	0.328600	0.646668	0.877641	07:32
3	0.268390	0.313912	0.659069	0.885380	07:42
4	0.289146	0.315030	0.664425	0.881406	07:53
5	0.259674	0.311650	0.698330	0.886216	07:59
6	0.254684	0.314927	0.684535	0.885798	07:57
7	0.240980	0.301015	0.699152	0.892909	07:58
8	0.220972	0.307936	0.712465	0.894374	08:01
9	0.193946	0.295319	0.729908	0.896883	07:59
10	0.190177	0.319130	0.710026	0.891864	08:02

After some changes to hyperparameters and oversampling, the following results have been achieved on 2 efficient nets. Theoretically bigger efficient nets such as Efficient Net B4 or

Efficient Net B7 would give better results, however, due to hardware limitation only the two models were trained. The results are seen below.

MODEL	ACC / F1
EfficietNet-b0	0.914 / 0.868
EfficietNet-b1	0.909 / 0.852

ResNeXt26 and ResNeXt50

The two models were chosen for study based on their good results on Image Net compared to other state-of-the-art models [31], as well as their usage in some of the researched papers. Most of the efficient nets provide a better accuracy in the same time being smaller than the alternatives. As the model is going to be used inside of an application, the size of it is very important. Below are the results of each model:

MODEL	ACC / F1
ResNeXt26	0.890 / 0.834
ResNeXt50	0.912 / 0.879

Hyperparemeters

In all cases the range of the learning rates was selected with the same technique of one cycle policy [32]. The graph of the learning rates and values of loss function were plotten correspondingly and the range of learning rates was chosen in the slice where the loss decreases in the most steepest way. In the case of the graph below the range of learning rates is selected to be between 1e-6 and 8e-6.

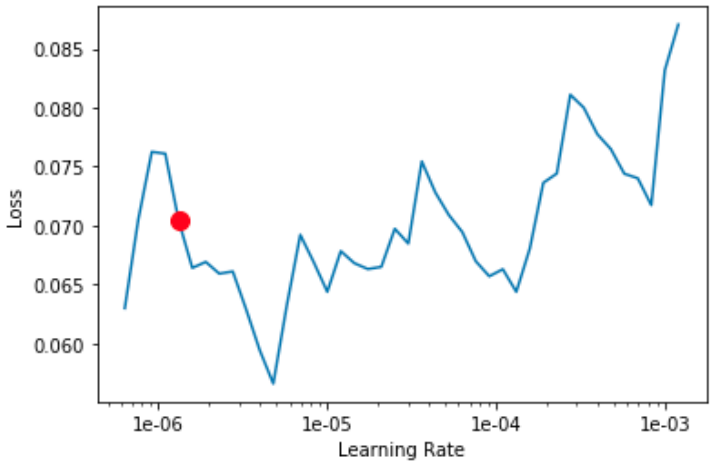


Figure 17. Learning Rate Slices

For the ADAM optimizer the recommended [40][41][42] hyperparameters were chosen as beta1 = 0.9 and beta2 = 0.99.

Oversampling

The experiments can be divided into two major parts - training on data without oversampling and with oversampling. Experiments on the oversampled and not oversampled dataset showed expected results. The accuracies of all models in both cases were similar (with an

error of 1%), however an increase in F1 score was substantial (more than 10%) in case of oversampled data.

Metrics

Two metrics were used for evaluation of the models - ***Accuracy*** *and* ***F1 score with macro averaging***. Accuracy is used, in order to compare the result to the accuracy of doctors (89%). Additionally, Macro averaged F1 score is calculated in order to understand the performance of the final solution on all classes.

Ensemble Learning

After getting promising results from most of the above described models, solving the class imbalance problem through oversampling, choosing the releavan metrics and hyperparameters, it was decided to create a new model by ensembling the EfficientNet-b0, EfficientNet-b1, ResNeXt26 and ResNeXt50 models in order to maximize the accuracy. The EfficientNet-b0, EfficientNet-b1, ResNeXt26 and ResNeXt50 were chosen as they showed an accuracy which was higher than the accuracy that doctors provide (89%) (In case of ResNeXt26 the accuracy was the same).

MODEL	ACC / F1
ResNeXt26	0.890 / 0.834
ResNeXt50	0.912 / 0.879
EfficietNet-b0	0.914 / 0.868
EfficietNet-b1	0.909 / 0.852

Table illustration of the models which showed the best accuracy and f1 score on the datasets.

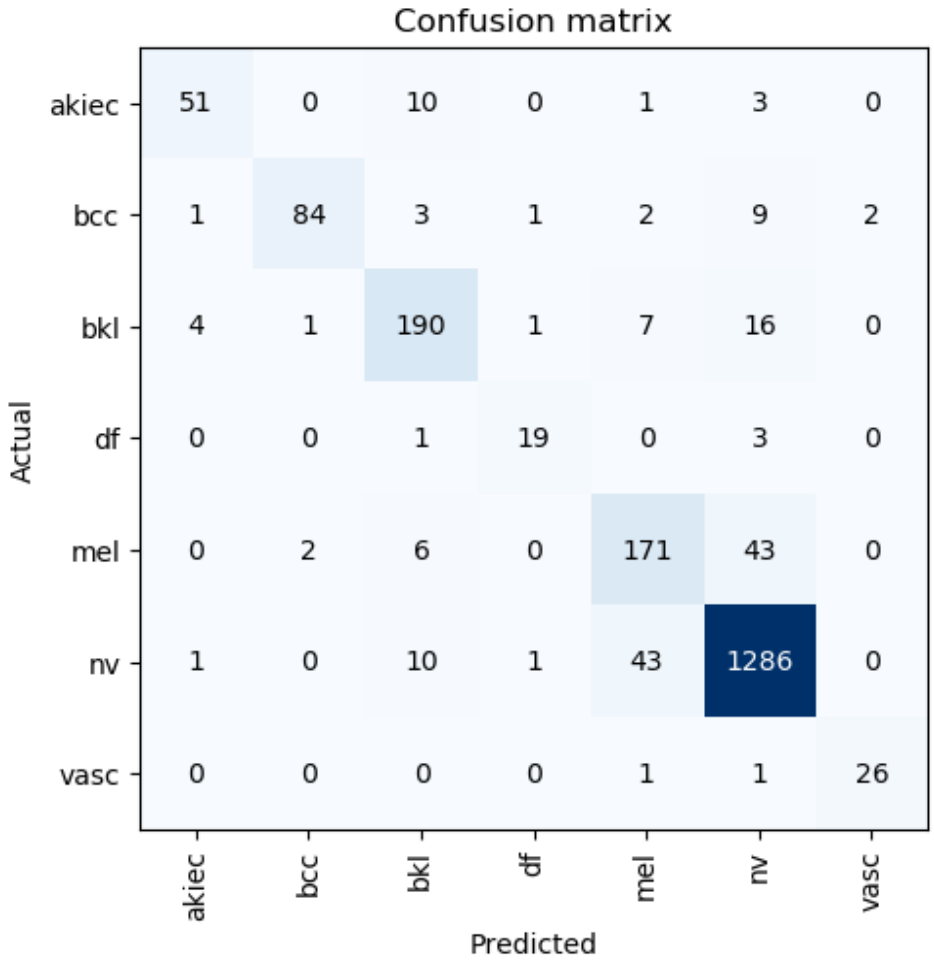


Figure 18. Confusion Matrix of the Ensembled model

The final ensemble model achieved the highest scores compared to the individual models. The achieved accuracy was 0.9275 and the macro-averaged F1 score was 0.9011. The confusion matrix of the ensemble model is presented above. Below is presented the comparison of the number of parameters in each model and their accuracy.

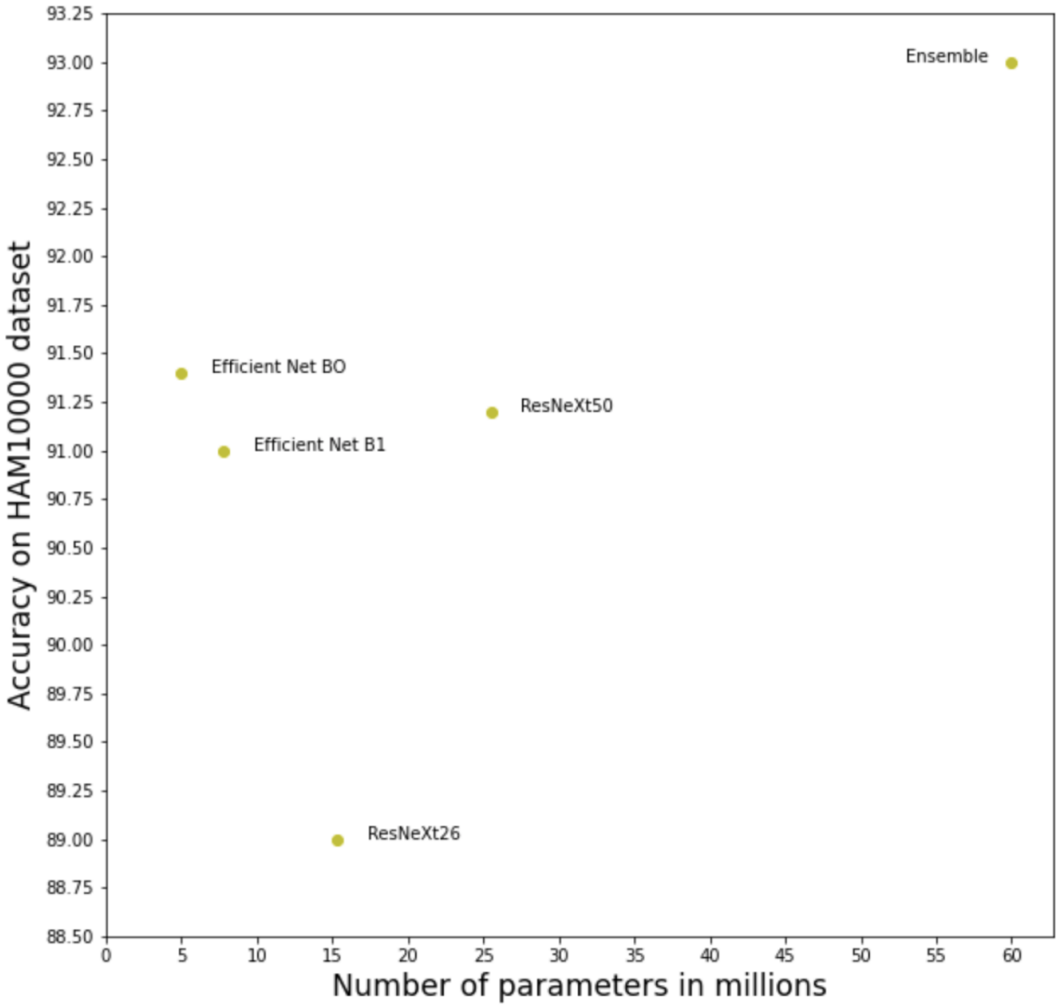


Figure 19. Comparison of the best performing state-of-the-art models and of the ensemble model

Mobile Application

After obtaining good results on solving skin cancer classification problem, I have decided to show a real life example on how the classification model could be used. An iOS application has been created to demonstrate the real life application of the problem.

Figure 20. Melanoma - Proposed App Name and Logo

The application acts as a tool for users to scan their skin for possible melanoma detection. After taking a picture frame with application, the application will use the already trained model to analyse the taken picture and provide an outcome as a probability of the mole being malignant, benign, etc.

The application will have 4 tabs: Scan, History, Summary and Account.

In account tab users will see their personal information and will be able to change it.

In history tab users will see the history of all the pictures that they took with the prediction.

In summary tab users will see their scan summary, important health reports and filters for date selection.

In the scan tab users will be able to analyze their skin using the already trained Deep Learning Convolutional Neural Network. This is the most important part of the application as it is here that the resuled model is used. In the scan tab the always on camera will tunnel each of the frames (image) through the model and will give the diagnosis and the probability of the diagnosis immediately with almost no delay.

The application will welcome users with a first one time login/registration page.

Below you can find several screenshots from the application:

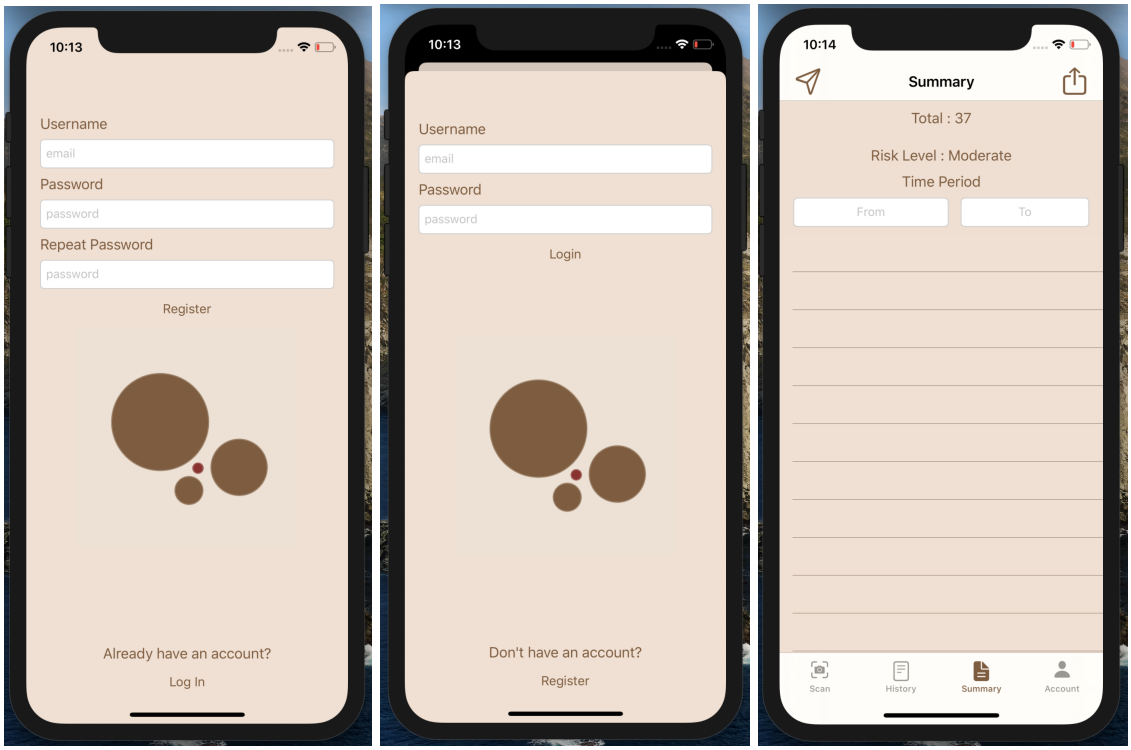
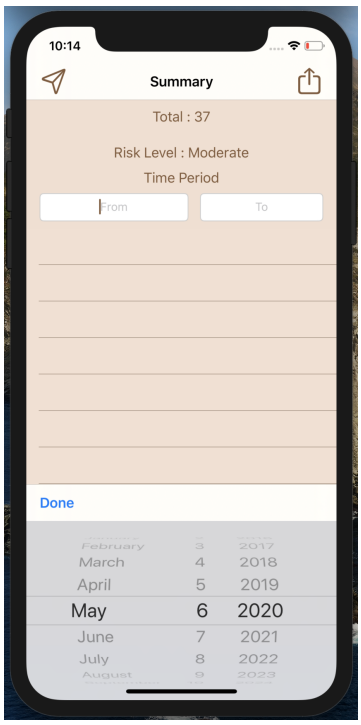


Figure 21. Figure 22. Figure 23. Screenshots from the application



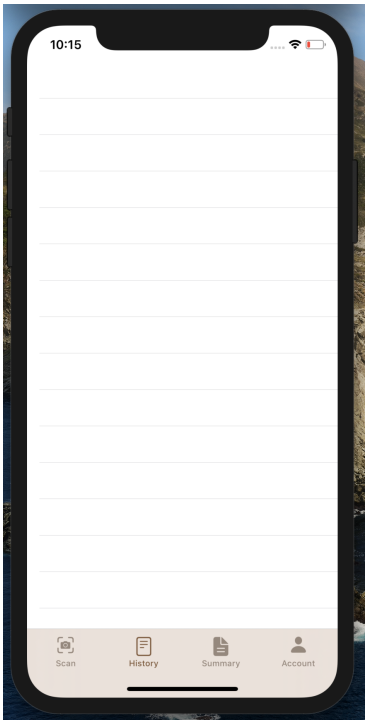
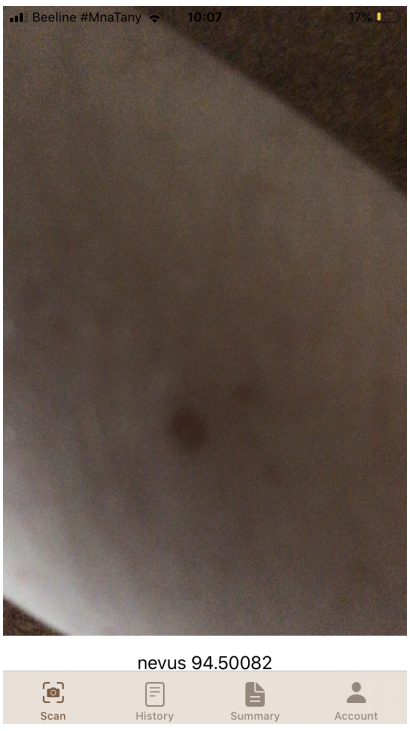


Figure 24. Summary Tab. Figure 25. History Tab. Figure 26. Scan Tab



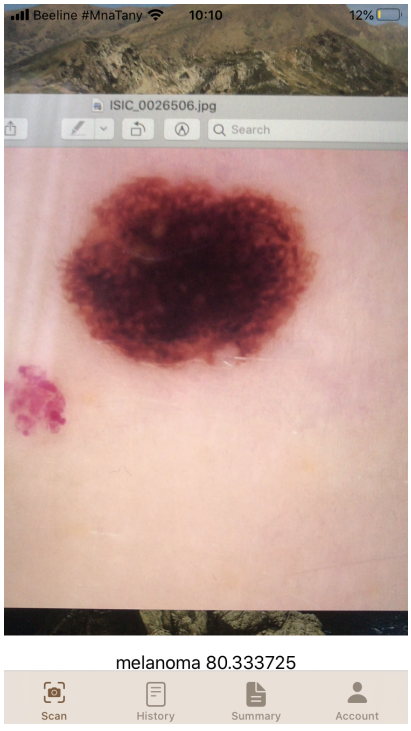


Figure 27. Figure 28. Live demonstration of the regular mole and skin cancer detection

Model Choice Options for the application

PyTorch

Although the ensemble of several models gave the best accuracy, the size of the application would increase several times and more computational power and resources would be required from the running device, because of the usage of several models. Therefore it is beneficial to use the Efficient Net B0 as a model in the application. Efficient Net B0 had a 91% accuracy compared to the 92% of the ensemble model.

PyTorch/FastAI to MLMODEL

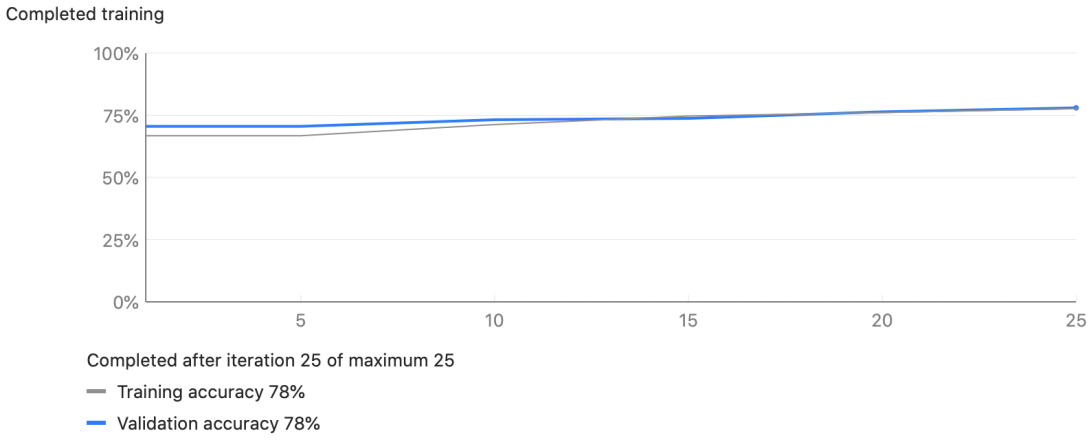
In order to use the Fastai model inside of an iOS application, it should be converted to Core MLMODEL file format. Conversion takes place in several stages:

Pytorch → ONNX → Apple Core ML

A detailed information on the conversion could be found in an article written by Hung Nguyen [33], where the stages of the conversion are shown in a step by step guide. If the model is a fastai model, then before the training several classes should be replaced to make it pytorch compatible. Then by using a converter the trained model can be converted to a pytorch model. Afterwards it can be then converted to onnx file format. And only after that using core ml tools the model could be converted to MLMODEL [34] file format which can be directly imported into an iOS application. A dummy application usage is described in an article written by Jianchao Li [35].

Create ML

If simplicity and performance are the highest priority then usage of the Create ML tool could also be possible. Create ML[36] is a framework which helps to create a classifier which can give a moderate accuracy with a simple model.



Class	Item Count	Precision	Recall
actinic keratosis	2	0%	0%
basal cell carcinoma	24	55%	50%
dermatofibroma	2	0%	0%
melanoma	38	45%	39%
nevus	266	87%	94%
pigmented benign keratosis	29	50%	45%
squamous cell carcinoma	9	40%	22%
vascular lesion	7	75%	43%

Figure 28. Create ML Results.

Ensemble

By using the backend server described in the section Backend Server it would also be possible to run the image taken by the mobile application through the whole ensemble of several

models, thus preserving the accuracy and not requiring computational resources from the mobile device.

Website, Backend Server and Database

Another live application of the model could be a website which will be created during future works where skin cancer/lesions would be analysed online.

Backend Server

Backend server will be created for future use of ensemble of models in the application. The backend server will also act as a tool for the application registration and data storage process.

Database

A database will be created using one of the variants:

- ORACLE DB
- MySQL
- MongoDB
- MariaDB

This will be a single database for website and application usage.

Tables

Illustration of the basic tables that will be necessary for the user registration and data accumulation.

User:

id	token	first_name	last_name	email	password

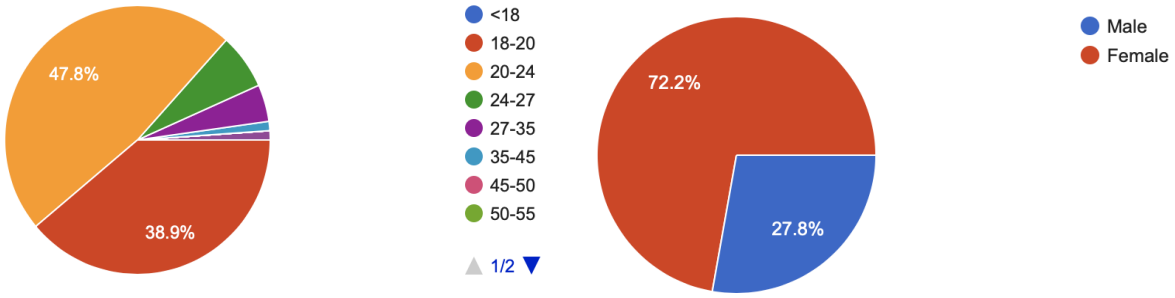
User_data:

id	user_id	image_id	diagnosis_1	diagnosis_2

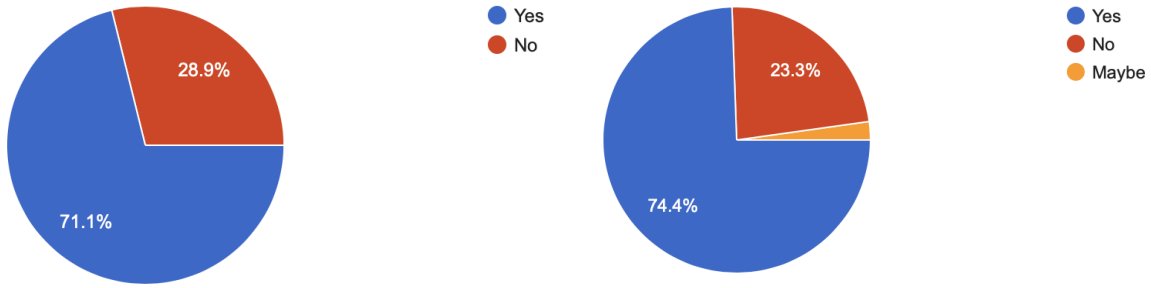
Survey

A survey has been conducted [37] in the frames of this project to understand the general public opinion about the project. Around 100 people have participated from all over the world.

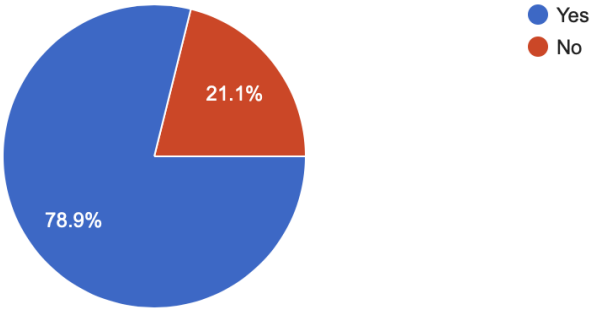
Age and Gender:



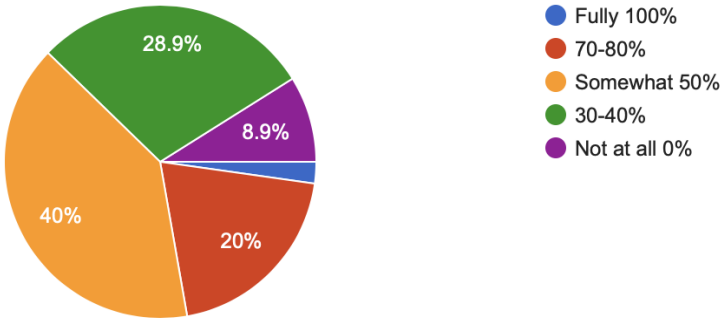
71% of respondents would use such an app and 74% would agree to share their data. Thanks to that more data will be accumulated over time for improving the Model.



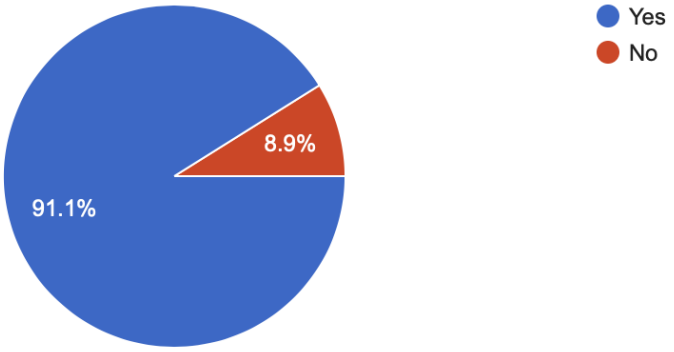
79.8% of people would recommend such an app to elderly. This means that elderly group may be a target audience for such an app.



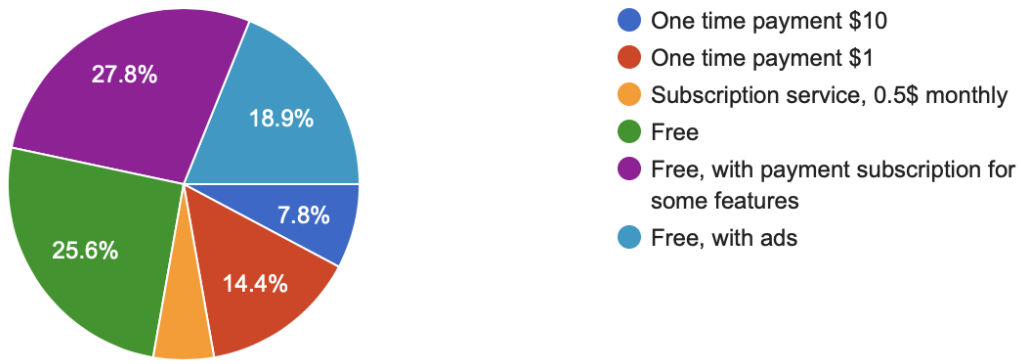
This is how much people would trust such an application. This means that a trust must be built before users will safely use the app. Maybe this can be done by getting some medical certification.



More than 90% of people would visit a doctor if the app tells them that they have a melanoma, which is very promising.



This shows how people would like to be charged for such an app. This means that such an app will mostly be successful being a free application with subscription services as well as providing a more professional paid option.



Target audience of the application are intended to be doctors (mainly dermatologists) and regular people who are curious about their moles as well. Based on the survey results we see that people from different age groups would prefer to use such an app. Moreover an interview has been conducted with several dermatologists in Armenia, and all of the respondents told me that they would be happy to have such an app even if it had a monthly subscription of $1, as it will save much time for them and act as a great tool for skin cancer detection.

Data Accumulation

With the help of the Application and Website in the future it is planned to gather new images of skin lesions. Case by case analyses will be done by doctors to provide an almost sure diagnosis. The gathered images will be used to increase the amount of the data that is used for the training of the models.

Conclusion and Future Work

The overall achieved result surpassess the results of skilled dermatologists by a significant amount. The model accurately classifies skin lesions of 7 classes on HAM-10000 Dataset and 10 classes on ISIC Archive Dataset.

There were multiple types of skin lesions, however the number of classes in the current model is limited based on the number of classes available in the current dataset, as well as the amount of available pictures in each class.

There were hundreds of approaches to the problem of skin cancer classification, where most of them were using either ensembles of several state-of-the-art models or just single state-of-the-art models.

Training was done on 2 datasets: ISIC Archive and HAM 10000 for comparing the results in between them and the final results were achieved by using only the HAM 10000 Dataset.

Several problems were encountered during the research and development, from the download of the dataset, image sizes, image quality, skin types, overfitting, model capacity to class imbalance. Those problems were solved by applying creative, well known, rare as well as new solutions.

During the research a base model was created for testing purposes, after which several well known state-of-the-art models were trained on the dataset.

The ensembled model of the trained state-of-the-art models was constructed and transformed to Core ML Model so it could be used in iOS devices.

A website and a backend server will be created in future works to make use of the created model so people can perform skin cancer analyses online. The backend server will also act as a tool for the application registration and data storage process.

A mobile application is created so regular people as well as doctors can perform skin cancer analyses with their devices.

The final result was achieved by using an Ensemble of Multiple Deep Transfer Learning Convolutional Neural Networks, specifically: EfficientNet-b0, EfficientNet-b1, ResNeXt26 and ResNeXt50. These were the models that had the highest performance during the training. The ensemble of those models gave the following results:

Accuracy 92.75 %

F1 score of 90.11 %.

In the future with the help of more powerful hardware, more time and a bigger team, it is planned to:

- Search for a bigger dataset and accumulation of new images,
- Increasing the number of classes,
- Trying different methods of ensemble learning,
- Using different models in ensemble learning,
- Using bigger versions of ResNet, EfficientNet and other models
- Making a multi race dataset and allocate a model for the dataset of each race.
- Creation of the website and backend server described in sections Website and Backend Server

- Finalization of the application described in section Mobile Application and entering the medical market
- Training of a simple model with the help of Create ML
- Use of a separate model for mole detection and feeding the augmented images in a training with new models
- Creation of a new model specifically designed for skin lesion and other human disease detection

Appendix

* Accuracy = # of correct predictions / # of all predictions

** The balanced accuracy in multiclass classification problems is a metric for dealing with imbalanced datasets. It is defined as the average of recall obtained on each class [38].

*** Macro-averaged F1-score [26]

Precision of a single class *'A'* shows the proportion of correctly predicted *'A'* cases in the total number of *'A'* predictions.

precision('A') = # of correct *'A'* predictions / # of all *'A'* predictions

Same for the recall:

recall('A') = # of correct *'A'* predictions / # of actual *'A'* in dataset

F1 by using the precision and recall:

F1('A') = 2 * precision('A') * recall('A') / (precision('A') + recall('A'))

Macro-averaged F1-score:

F1-macro = (F1('class 1') + … + F1('class n')) / n

References

[1] GBD. (2015). Disease and Injury Incidence and Prevalence, Collaborators. (8 October 2016). "Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990–2015: a systematic analysis for the Global Burden of Disease Study 2015". Lancet. 388 (10053): 1545–1602. doi:10.1016/S0140-6736(16)31678-6. PMC 5055577. PMID 27733282.

[2] GBD 2015 Mortality and Causes of Death, Collaborators. (8 October 2016). "Global, regional, and national life expectancy, all-cause mortality, and cause-specific mortality for 249 causes of death, 1980–2015: a systematic analysis for the Global Burden of Disease Study 2015". Lancet. 388 (10053): 1459–1544. doi:10.1016/s0140-6736(16)31012-1. PMC 5388903. PMID 27733281.

[3] V J Mar, H P Soyer. Artificial intelligence for melanoma diagnosis: How can we deliver on the promise? Annals of Oncology, (2018); DOI: 10.1093/annonc/mdy193

[4] H A Haenssle, C Fink, R Schneiderbauer, F Toberer, T Buhl, A Blum, A Kalloo, A Ben Hadj Hassen, L Thomas, A Enk, L Uhlmann. Man against machine: diagnostic performance of a deep learning convolutional neural network for dermoscopic melanoma recognition in comparison to 58 dermatologists. Annals of Oncology, (2018); DOI: 10.1093/annonc/mdy166

[5] Skin Cancer Foundation, (2020), Retrieved from https://www.skincancer.org/skin-cancer-information/melanoma/melanoma-warning-signs-and-images/

[6] Giorgio Giacinto, Fabio Roli. 2001. “Design of effective neural network ensembles for image classification purposes” Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy. Retrieved from https://doi.org/10.1016/S0262-8856(01)00045-2

[7] A. Kumar, J. Kim, D. Lyndon, M. Fulham and D. Feng, "An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification," in IEEE Journal of Biomedical and Health Informatics, vol. 21, no. 1, pp. 31-40, Jan. 2017, doi: 10.1109/JBHI.2016.2635663.

[8] Dongmei Han, Qigang Liu, Weiguo Fan. (2018). “A new image classification method using CNN transfer learning and web data augmentation” Retrieved from https://doi.org/10.1016/j.eswa.2017.11.028

[9] Skin Lesion Analysis Towards Melanoma Detection. (2019). https://challenge2019.isic-archive.com

[10] Yun Jiang, Li Chen, Hai Zhang, Xiao Xiao. (2019) “Breast cancer histopathological image classification using convolutional neural networks with small SE-ResNet module”. Retrieved from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6440620/

[11] Andre Esteva, Brett Kuprel, Roberto A. Novoa, Justin Ko, Susan M. Swetter, Helen M. Blau & Sebastian Thrun. Dermatologist-level classification of skin cancer with deep neural networks, (2017), p. 116, Macmillan Publishers Limited

[12] Mengting Xu, Tao Zhang, and Zhongnian Li. (2019). “VGG-16 for Skin Lesion Diagnosis: Participation in ISIC 2019 Skin Lesion Classification Challenge” Nanjing University of Aeronautics and Astronautics, JiangSu, China. Retrieved from https://isic-challenge-stade.s3.amazonaws.com/eadacb20-a2d8-4fc5-9362-02ed287bbac0/isic2019_manuscript_task1.pdf?AWSAccessKeyId=AKIA2FPBP3II4S6KTWEU&Signature=X%2F7sjA077sDTwAyNxzEessIzaQI%3D&Expires=1589377554

[13] Anand Nambisan. (2019). “Deep Learning Approach for the ISIC 2019 Challenge”. Retrieved from https://isic-challenge-stade.s3.amazonaws.com/842d89cb-221b-4c1b-a938-47008e2b8150/isci2019_sumbission.pdf?AWSAccessKeyId=AKIA2FPBP3II4S6KTWEU&Signature=cr78UMXpZeHWTbYZjXnb4aVBPSU%3D&Expires=1589296323

[14] ISIC Challenge. (2019). https://challenge2019.isic-archive.com/leaderboard.html (Link)

[15] N. C. F. Codella et al., "Deep learning ensembles for melanoma recognition in dermoscopy images," in IBM Journal of Research and Development, vol. 61, no. 4/5, pp. 5:1-5:15, 1 July-Sept. 2017, doi: 10.1147/JRD.2017.2708299.

[16] Nils Gessert, Maximilian Nielsen, Mohsin Shaikh, René Werner, and Alexander Schlaefer. (2019). “Skin Lesion Classification Using Loss Balancing and Ensembles of Multi-Resolution EfficientNets”. Institute of Medical Technology, Hamburg University of Technology, Hamburg,

Germany.Retrieved from https://isic-challenge-stade.s3.amazonaws.com/99bdfa5c-4b6b-4c3c-94c0-f614e6a05bc4/method_description.pdf?AWSAccessKeyId=AKIA2FPBP3II4S6KTWEU&Signature=qVp57G3qtx9%2Bc2yvVxWF0QXNchU%3D&Expires=1587737348 (Link)
https://github.com/ngessert/isic2019 (source code)

[17] Steven Zhou , Yixin Zhuang, Rusong Meng. (2019). "Multi-Category Skin Lesion Diagnosis Using Dermoscopy Images and Deep CNN Ensembles". Retrieved from https://isic-challenge-stade.s3.amazonaws.com/9e2e7c9c-480c-48dc-a452-c1dd577cc2b2/ISIC2019-paper-0816.pdf?AWSAccessKeyId=AKIA2FPBP3II4S6KTWEU&Signature=wcJVX6Ed7i5kdg1wCJZFjc7oN0I%3D&Expires=1588207694 (Pdf)

[18] Ashraful Alam Milton. (2018). "Automated Skin Lesion Classification Using Ensemble of Deep Neural Networks in ISIC 2018: Skin Lesion Analysis Towards Melanoma Detection Challenge". Retrieved from https://arxiv.org/pdf/1901.10802.pdf (Pdf)

[19] ISIC Archive, (2020), Retrieved from https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/gallery

[20] Tschandl, Philipp, (2018), "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions", Retrieved from https://doi.org/10.7910/DVN/DBW86T, Harvard Dataverse, V1

[21] Mateusz Buda Atsuto Maki, Maciej A. Mazurowski (2018). "A systematic study of the class imbalance problem in convolutional neural networks" . Retrieved from https://arxiv.org/pdf/1710.05381.pdf (PDF)

[22] Sargsyan, D. (2020). armisicdownloader.py

[23] ISIC Archive API Documentation, (2020), Retrieved from https://www.isic-archive.com/#!/topWithHeader/onlyHeaderTop/apiDocumentation

[24] F Beta Score. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.fbeta_score.html

[25] Balanced Multi class accuracy. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.metrics.fbeta_score.html

[26] Dell Zhang, Jun Wang, Xiaoxue Zhao. (2015). “Estimating the Uncertainty of Average F1 Scores”. ICTIR '15: Proceedings of the 2015 International Conference on The Theory of Information RetrievalSeptember 2015 Pages 317–320 https://doi.org/10.1145/2808194.2809488

[27] Brownlee, J. (2020). “SMOTE for Imbalanced Classification with Python”. Retrieved from https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[28] Opitz, D.; Maclin, R. (1999). "Popular ensemble methods: An empirical study". Journal of Artificial Intelligence Research. 11: 169–198. doi:10.1613/jair.614

[29] Polikar, R. (2006). "Ensemble based systems in decision making". IEEE Circuits and Systems Magazine. 6 (3): 21–45. doi:10.1109/MCAS.2006.1688199

[30] Rokach, L. (2010). "Ensemble-based classifiers". Artificial Intelligence Review. 33 (1–2): 1–39. doi:10.1007/s10462-009-9124-7

[31] Mingxing Tan, Staff Software Engineer and Quoc V. Le. (2019). “EfficientNet: Improving Accuracy and Efficiency through AutoML and Model Scaling”. Retrieved from https://ai.googleblog.com/2019/05/efficientnet-improving-accuracy-and.html. Google AI

[32] Leslie N.Smith (2018). A disciplined approach to neural network hyper-parameters: Part 1 -- learning rate, batch size, momentum, and weight decay

[33] Nguyen Minh H. (2018) “Convert fast.ai trained image classification model to iOS app via ONNX and Apple Core ML “ Retrieved from https://medium.com/@hungminhnguyen/convert-fast-ai-trained-image-classification-model-to-ios-app-via-onnx-and-apple-core-ml-5fdb612379f1

[34] Apple Inc. (2019). Core ML Model Format Specification. Retreived from https://apple.github.io/coremltools/coremlspecification/

[35] Jianchao Li.”From PyTorch to Core ML”. (2019). Retrieved from https://jianchao-li.github.io/post/from-pytorch-to-coreml/

[36] Apple Inc. Create ML. Retrieved from https://developer.apple.com/videos/play/wwdc2018/703/

[37] Sargsyan, D. (2020). Melanoma Detection and Skin Cancer Classification App - Survey, (2020). Retrieved from https://forms.gle/fFQpdbDZ7VJ2WLok8

[38] D. R. Velez, B. C. White, A. A. Motsinger, W. S. Bush, M. D. Ritchie, S. M. Williams, et al., "A balanced accuracy function for epistasis modeling in imbalanced datasets using multifactor dimensionality reduction", *Genetic Epidemiology*, vol. 31, no. 4, pp. 306-315, May 2007.

[39] Avineri, G. (2019). "ISIC Archive Downloader" Retrieved from https://github.com/GalAvineri/ISIC-Archive-Downloader

[40] L´eon Bottou. (2012) "Stochastic Gradient Descent Tricks" Retrieved from https://www.microsoft.com/en-us/research/wp-content/uploads/2012/01/tricks-2012.pdf

[41] Yoshua Bengio. (2012). "Practical Recommendations for Gradient-Based Training of Deep Architectures". Retrieved from https://arxiv.org/pdf/1206.5533v2.pdf

[42] Diederik P. Kingma, Jimmy Ba. "Adam: A Method for Stochastic Optimization". 3rd International Conference for Learning Representations, San Diego, 2015. Retrieved from https://arxiv.org/abs/1412.6980

[43] Shin Ando, Chun Yuan Huang. (2017). "Deep Over-sampling Framework for Classifying Imbalanced Data". School of Management Tokyo University of Science Chiyoda-ku Japan Retrieved from https://link.springer.com/chapter/10.1007/978-3-319-71249-9_46

[44] Brownlee, J. (2020) "Undersampling Algorithms for Imbalanced Classification" Retrieved from https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/