University of Padua

Department of Computer Science

Skin cancer classification using Keras

Alberto Bezzon 1211016

Tommaso Carraro 1210937

The report resumes briefly the cognitive services project

Academic year 2018 - 2019

1 Introduction

Skin cancer is the most widespread cancer and one of the most dangerous because of the number of cases that not only exceeds the combined total of new cases for prostate cancer, breast cancer, lung cancer, and colorectal cancer, but also it increases from year to year. Malignant melanoma is a prevalent type of cancer that is especially deadly. It is well known that early detection and proper treatments for new malignant skin cancer cases are very important to ensure high survival rate. Indeed, with an appropriate treatment in an early stage, survival rates are very promising. Otherwise, the survival rate for melanoma decreases from 99% to 14% in more advanced stages.

The usually way to detect a melanoma is by inspecting the visual details of skin which has a low precision. Another way is dermoscopy, a non-invasive technique, that can capture a high resolution image of the skin which enables dermatologists to detect features which are invisible to the naked eye. This technique makes easier to diagnose melanoma, but it is time consuming and it is based on the skill of clinician that made dermoscopy. Moreover, because of the resemblance between malignant skin tumors and benign skin lesions in visual features, it is very hard for dermatologists to differentiate between them.

In recent years, deep learning, and specifically convolutional neural networks (CNNs), have reached very good performance in skin cancer classification tasks and have allowed computers to outperform dermatologists. For this reason, in our project we have tried to improve the accuracy of skin cancer detection using state-of-the-art CNN models and techniques. In particular, we have used these models to distinguish between seven common types of skin cancers that are included in the HAM10000, a recent and famous dataset made for this specific task. We obtained the best performance on the test set using data augmentation to train a simple CNN model built from scratch.

This document is organized as follows:

- Section 2 presents related works for "Skin cancer classification";
- Section 3 provides the description of dataset used in our experiments;
- Section 4 includes the approach used for solving the task;
- Section 5 presents the experiments we made;
- Section 6 contains the conclusion, the results of experiments and future works.

2 Related works

We have viewed different state-of-the-art papers. In [1] researchers used Googles Inception v3 CNN architecture pretrained on the 2014 ImageNet Challenge. They then removed the final classification layer from the network and retrained it with their dataset, fine-tuning the parameters across all layers. During training they resized each image to 299 299 pixels in order to made it compatible with the original dimensions of the Inception v3 network architecture. All layers of the network were fine-tuned using the same global learning rate and RMSProp optimizer. They performed their experiments on a 129.000 images dataset,

created from a combination of different datasets. They obtained 72.1% overall accuracy training their model on 757 classes. To create these training classes they used a taxonomy of skin disease and a partitioning algorithm that maps diseases into training classes. We have tried to obtain these data, but they were protected by the Stanford Hospital.

In [2] researchers studied the effectiveness and capability of different pre-trained state-of-the-art CNN architectures (DenseNet 201, ResNet 152, Inception v3, InceptionResNet v2). All the models they used were pre-trained on the 2014 ImageNet Challenge. They changed the classification part of these models with a custom classifier and they retrained them across all the layers using different hyperparameters depending on the specific network architecture. They trained these models on a dataset composed of 10.135 dermoscopy skin images composed by the combination of HAM10000 and PH2 datasets. The aim of their project was to compare the ability of deep learning with the performance of highly trained dermatologists. Overall, the mean results show that all deep learning models outperformed dermatologists (at least 11%). The best ROC AUC values for melanoma and basal cell carcinoma are 94.40% (ResNet 152) and 99.30% (DenseNet 201) versus 82.26% and 88.82% of dermatologists, respectively.

In [3] researchers used a CNN model built from scratch. The proposed model architecture consists on a sequence of alternating Conv2D and MaxPooling2D layers that form the core building blocks of modern CNNs. Then, they putted a batch normalization layer after each ReLu activation. They used this model to perform a binary classification task, in fact the first convolutional layer takes in 224 x 224 skin lesion images and the last dense layer contains a single unit with sigmoid activation in order to output the resulting classes (benign and malignant). They trained their model on the PHDB melanoma dataset, created by their own from a combination of open access datasets. They obtained an accuracy of the 86% on this dataset and regularization techniques such as dropout (0.5) and data augmentation techniques were heavily relied upon to combat the overfitting problem.

After reading these papers we decided firstly to try the power of transfer learning on our dataset. Given the big amount of data in HAM10000 and the high difference between the images in our dataset and the images included in the ImageNet Challenge, the best transfer learning strategy was to train the pre-trained model across all the layers. This technique gives us bad performance compare to the training of a simple CNN model built from scratch. So, we decided to use an approach that is more similar to the method explained in [3] than the other papers.

3 Dataset

On the Internet there are few open access datasets for skin cancer classification task, and most of them contain a collection of bad quality images that are not biopsy proven. We read a lot of papers on this task and in most of them researchers had to create a dataset from scratch for the unavailability of a complete dataset composed of quality and biopsy proven images on the web. In most cases they took best images from different sources and they built their own dataset. Recently, some researchers and dermatologists understood the importance of this task and decided to create a new dataset, called HAM10000[4], that we used on our project. This dataset contains 10015 dermatoscopic images that were collected over a period of 20 years from two different sites, the Department of Dermatology at the Medical University of Vienna, Austria, and the skin cancer practice of Cliff Rosendahl in Queensland, Australia.

It includes pigmented lesions from different populations. The Austrian image set consists of lesions of patients referred to a tertiary European referral center specialized for early detection of melanoma in high risk groups. The Australian image set includes lesions from patients of a primary care facility in a high skin cancer incidence area. Dermatoscopic images of both study sites were taken by different devices using polarized and non-polarized dermatoscopy. The set includes representative examples of pigmented skin lesions that are practically relevant. More than 95% of all lesion encountered during clinical practice will fall into one of the seven diagnostic categories contained in the dataset. In practice, the task of the clinician is to differentiate between malignant and benign lesions, but also to make specific diagnoses because different malignant lesions may be treated in a different way and timeframe. The number of images in the datasets does not correspond to the number of unique lesions, because experts also provide images of the same lesion taken at different magnifications or angles, or with different cameras. This should serve as a natural data-augmentation as it shows random transformations and visualizes both general and local features.

The seven different categories of skin lesions contained in the dataset are:

akiec Actinic Keratoses (Solar Keratoses) and Intraepithelial Carcinoma (Bowens disease) are common non-invasive, variants of squamous cell carcinoma that can be treated locally without surgery. There is agreement that these lesions may progress to invasive squamous cell carcinoma.

bcc Basal cell carcinoma is a common variant of epithelial skin cancer that rarely metastasizes but grows destructively if untreated.

bkl "Benign keratosis" is a generic class that includes seborrheic keratoses ("senile wart"), solar lentigo and lichen-planus like keratoses (LPLK), which corresponds to a seborrheic keratosis or a solar lentigo with inflammation and regression. From a dermatoscopic view, lichen planus-like keratoses are especially challenging because they can show morphologic features mimicking melanoma and are often biopsied or excised for diagnostic reasons.

df Dermatofibroma is a benign skin lesion regarded as either a benign proliferation or an inflammatory reaction to minimal trauma.

nv Melanocytic nevi are benign neoplasms of melanocytes and appear in a myriad of variants, which all are included in this dataset. The variants may differ significantly from a dermatoscopic point of view.

mel Melanoma is a malignant neoplasm derived from melanocytes that may appear in different variants. If excised in an early stage it can be cured by simple surgical excision.

vasc Vascular skin lesions in the dataset range from cherry angiomas to angiokeratomas and pyogenic granulomas. Hemorrhage is also included in this category.

In Figure 1 is possible to see the difference between these types of skin lesions.

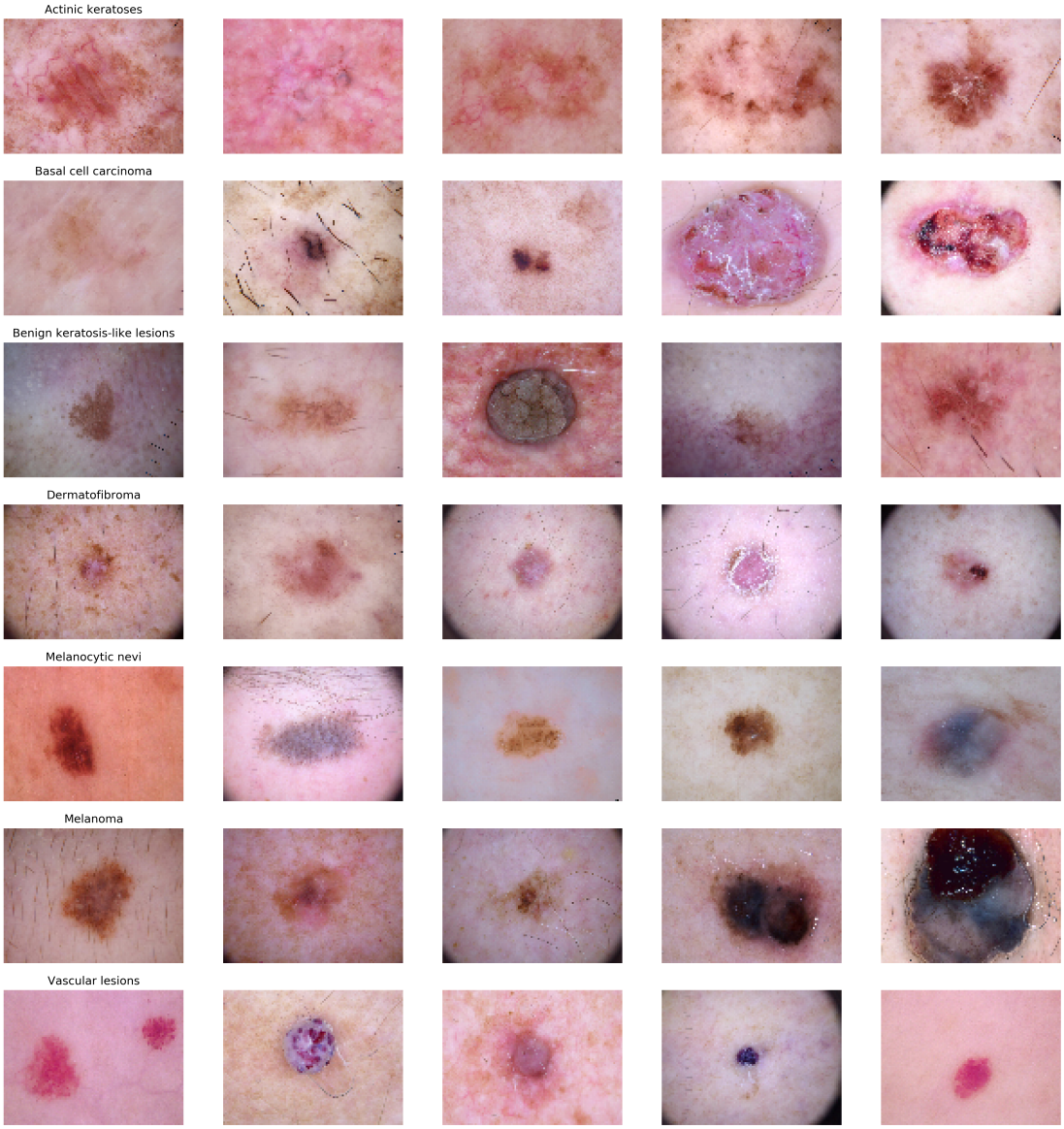


Figure 1: *Categories of skin lesions in HAM10000*

Unbalanced data is the biggest problem of HAM10000, in fact melanocytic nevi is the majority class with 67% of examples, instead dermatofibroma is the minority class with only 4% of examples. In Figure 2 is possible to see the distribution of classes.

Unbalanced data is a common issue of skin cancer datasets and has been a challenge for this project. Initially we thought to find other images to oversample the dataset, but then we turned out that images in the internet are bad in quality and different compared to the images of the HAM10000, so we have decided to keep it at the original version. Data preprocessing in this dataset is limited to the normalization of images and the transformation of labels in one-hot encoded vectors to make the net capable to learn from them. We have decided

to split the dataset in 80% training set, 10% validation set and 10% test set. We kept this aggressive split because of the limited amount of data, in fact we want our model to learn as much as possible features from the dataset. Images have been resized to 100 x 75 before being supplied to the CNN model. We have decided this specific sizes because experiments showed that the model works better with small resolution images.

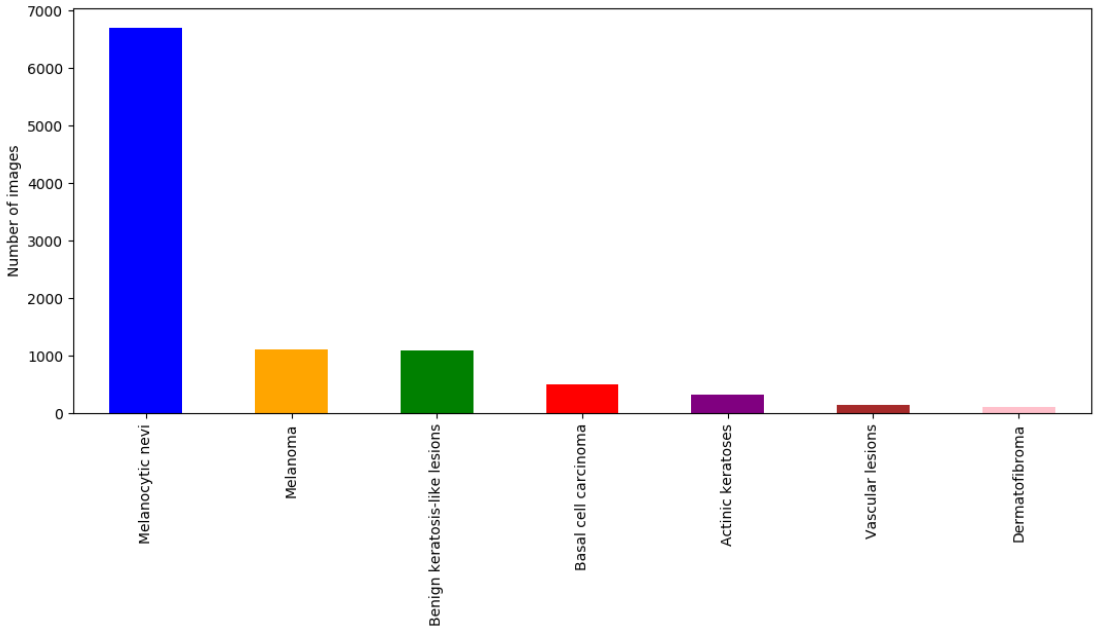


Figure 2: *Distribution of categories in HAM10000*

4 Proposed model

4.1 Performance measures choice

As mentioned in Section 3 HAM10000 is a highly imbalanced dataset and it has been a challenge to deal with it. The first problem we faced was on the choice of the right performance measure for our experiments. Accuracy is usually referred as the best performance measure for image classification tasks but in case of imbalanced data is definitely not the right choice. In fact, if the accuracy is high it does not mean that we have found a good model, because it can be possible that it is high only because our model is very good in predicting majority classes, but we want our model to be accurate on all the classes, even the most challenging minority classes.

Our goal is to obtain a model that maximize both precision and recall on all the classes. To achieve this goal, we have decided to use F1-measure and macro average performance measures. F1-measure allows to understand in which classes our model performs well: a high F1 score for one class means that our model returns a high precision but also a high recall on that specific class.

We have then selected the macro average F1 because we are interested in knowing if our model performs well in minority classes, which include malign skin cancers, such as melanoma

class, that is the most important one in the dataset. Moreover, if the model performs well in minority classes, even if the number of examples is limited, means that it is able to learn the correct features to discriminate between these classes.

Finally, since we are interested in predicting well malign skin cancers, we did not use micro average because it returns high scores when the model performs well in majority classes, which include benign tumors.

4.2 Transfer learning attempt

As mentioned in Section 2 we firstly tried to apply transfer learning on our task. We took Inception v3 model, pre-trained on ImageNet Challenge, and we then changed the classification part on top of it with a custom classifier. This classifier was composed of one global average pooling layer, two fully connected layers and finally a softmax layer. We took this architecture from [3] because they had obtained the best results we have found on HAM10000. We froze all the layers except of the custom classification part and we trained our model with the settings reported in [3]. We obtained bad performances with this model compared to the performance reported in the paper. The accuracy was relatively high (67%) but when we moved to the confusion matrix we discovered that the model was predicting everything like a Melanocytic Nevi to maximize the overall accuracy. So, the model wasn't able to learn anything. After this test we decided to move to the classic CNN architecture, in fact transfer learning performs well when the dataset is similar to the ImageNet dataset and it wasn't the case.

4.3 Architecture of proposed model

For the design of our CNN model architecture we have followed a common design pattern in CNN. This pattern consists in building a simple model architecture with only one convolutional block and to increase its depth until the adding of a new convolutional block does not give better performance compared to the last model built. We proceeded this way because we found that this pattern performs well in general image classification tasks and the experiments in [2] showed that it performs well even in skin cancer classification task. So, we began our project with the construction of a base model that we have then tried to improve. In Figure 3 we present our base model. Its architecture is composed of a convolutional block and a simple classification block, which are the building blocks of modern CNN architectures. The convolutional block is composed of two convolutional layers with 32 activation maps followed by a max pooling layer. We have decided to set a 3x3 filter size for the convolutional layers because local features in skin lesions are relatively small. The classification block is composed of a flatten layer used to flat the output of the convolutional block, a fully connected layer with 64 units and finally a softmax layer. We have then added a dropout layer after the fully connected layer. We will explain our regularizations techniques in section (regularization).

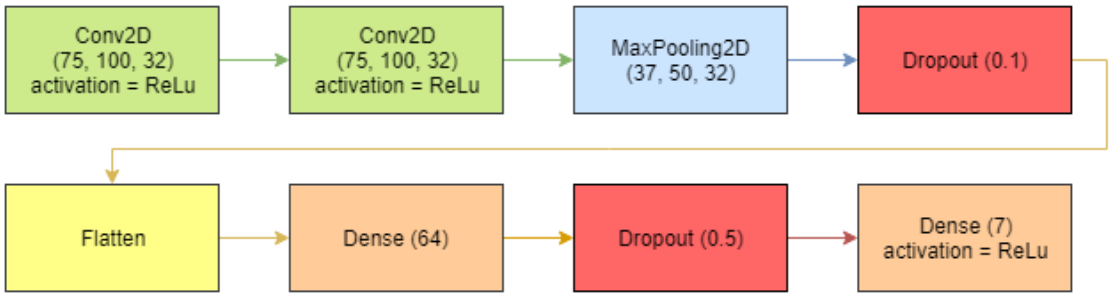


Figure 3: *Base model architecture*

Given the base model we began to add convolutional blocks to it to see if model performances increased. The adding of a convolutional block consists in add two convolutional layers with doubled activation maps followed by a max pooling layer after the previous convolutional block of the net. With this technique we have changed the architecture of our base model five times until we obtained our best and proposed model. In Table 1 is possible to compare the performances of the models we have tried. The table presents the F1 scores for all the classes, the macro average of the f1 score and the test accuracy for each model. We trained these models with the settings reported in Section 4.4.3.

Model	**akiec**	**bcc**	**bkl**	**df**	**mel**	**nv**	**vasc**	**Macro average**	**Test accuracy**
Base model[1]	0.32	0.45	0.47	0.13	0.34	0.87	0.49	0.44	0.74
Model 2[2]	0.40	0.53	0.50	0.00	0.36	0.88	0.72	0.48	0.76
Model 3[3]	0.38	0.52	0.52	0.07	0.47	0.89	0.58	0.49	0.77
Proposed model[4]	0.41	0.52	0.53	0.16	0.38	0.90	0.65	0.51	0.76
Model 5[5]	0.34	0.46	0.53	0.07	0.39	0.90	0.62	0.47	0.75

[1] Model with
2
3
4
5

Table 1: *Models performances*

The base model has the worst performances compared to the other models. Model 2 increases the performances of the base model on all the classes, but it has the problem to be bad in classifying Dermatofibromas. Model 3 has the best accuracy on the test set but a macro average that is less than our proposed model and we prefer to balance the number of correct classifications over the classes instead of increasing the overall accuracy. Even if model 3 has the best performance on melanoma detection we chose model 4 because we think it is able to capture more complex features on the images, in fact it has the highest F1 score in Dermatofibroma class, that seems to be the most difficult to classify for our models, and also it outperforms the other models on all the other classes. We think that our proposed model

can reached these performances thanks to the number of convolutional blocks on it, in fact more convolutional blocks a network has and more complex features on the images will be able to detect. Moreover, we observed that model 4 has reached the maximum depth for our CNN in this specific task, in fact the adding of another convolutional block decreases most of the metrics scores. Finally, every time we added a new convolutional block we doubled also the number of units in the fully connected layer at the top of the network. We chose this strategy because more complex are the features that the convolution block is able to identify and more units in the fully connected layer will be required to capture these complex features.

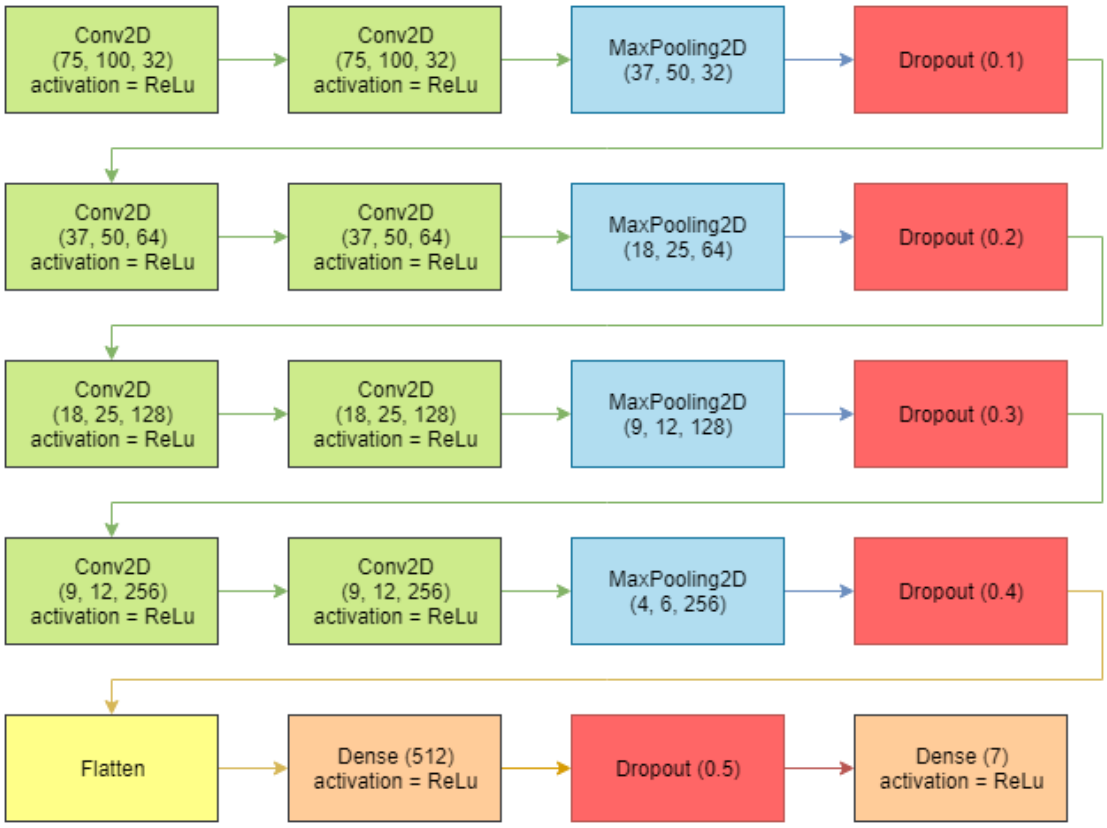


Figure 4: *Proposed model architecture*

In Figure 4 we present the proposed model architecture. It performs well on HAM10000 compared to transfer learning and it has the advantage of the simplicity in the architecture. It is less depth and it takes faster training times thanks to the smaller number of parameters to be learnt. After we have found the best architecture for this task we tried to do some experiments to increase its performances, such as class weighting and oversampling with data augmentation to deal with the problem of imbalanced data. In Section 5 we present these experiments.

4.4 Regularizations and training settings

4.4.1 Regularizations techniques

To prevent overfitting in our models we decided to use three common regularizations techniques:

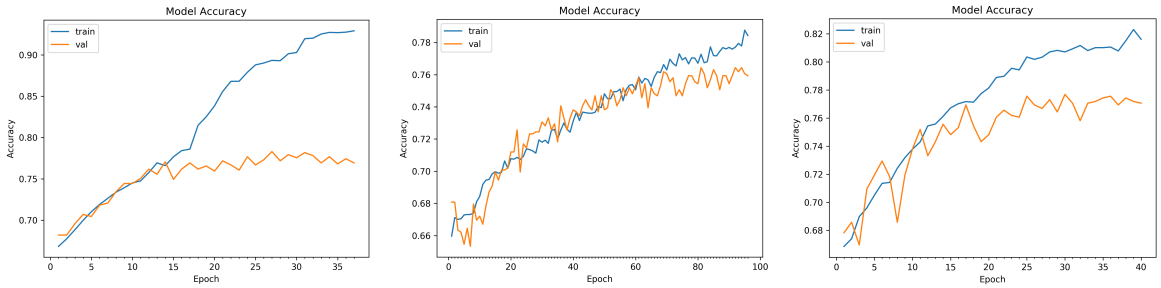
- **Dropout layers:** in previous section we presented our model architecture and it presents five dropout layers. The dropout layer after the fully connected layer freezes half of the units at each epoch. We set this percentage because it is well known that fully connected layers are highly subject to overfitting. We then decided to add a dropout layer after each convolutional block, with an increasing percentage of frozen units. Since the number of parameters to be learnt increases with the increase of the depth of the network this configuration helped our network to mitigate the overfitting issue;
- **Learning rate decay:** we set our model to decay the learning rate if no improvements on the validation accuracy are showed after 3 epochs of training. We tried many experiments and we discovered that this configuration helped the network avoids overfitting;
- **Early stopping:** since dropout and learning rate decay were not enough to avoid overfitting, early stopping was the next choice. Our models are configured to stop the learning in case of high overfitting. If the validation accuracy does not increase in 10 epochs, the training will be stopped and the weights corresponding to the best validation accuracy will be restored. We decided this configuration because we observed that in many cases the model needed many epochs to continue to learn.

4.4.2 Optimizer and optimizer hyperparameters choice

We have tried three optimizers for the gradient descent: Stochastic Gradient Descent, RMSprop and Adam optimizers. We limited the test to these three optimizers because we found they are commonly used in skin classification tasks. SGD and RMSprop did not perform well on our task. They did not allow the net to train, in fact the validation accuracy was stuck at 68%. We had the same problem of transfer learning. We then tried Adam optimizer and the network began to learn.

We have selected only the learning rate of the Adam optimizer and we left all the other parameters at their default. We tried to train our models with a 0.001 learning rate. We have found that is the common choice in computer vision, but in our case the learning was too fast and the network began to overfit soon. We tried 0.00001 learning rate and we observed that the convergence was very stable, in fact the training accuracy and the validation accuracy followed the same behavior, but models took too much epochs to converge and very often the early stopping stopped the training even if there was not overfitting because 10 epochs were not enough to improve the training accuracy.

Finally, we have selected 0.0001 learning rate because we observed it was the best tradeoff between convergence speed and overfitting. In Figure 5 we show how training and validation accuracy behaviours change with the change of the learning rate.



(a) *Learning rate 0.001*

(b) *Learning rate 0.00001*

(c) *Learning rate 0.0001*

Figure 5: *Validation accuracy plot with different learning rate*

4.4.3 Training settings

We have trained all our models with these settings:

- Adam optimizer with 0.0001 learning rate and the other optimizer parameters at their default;
- categorical cross entropy loss function;
- 32 batch size: this is the default batch size in keras and we observed good performances on our model with this batch size;
- 200 epochs: we decided this number to allow the network to learn as much as possible from the training set. If the model begins to overfit the early stopping will stop the training and it will restore the best weights;
- Regularization techniques showed in section 4.4.1.

5 Experiments

5.1 Experimental enviroments

Training a deep learning model that involves intensive compute tasks on large dataset can take days to run on a single CPU or a slow GPU. In our case, since HAM10000 dataset has 10015 images, it is unthinkable to perform the training of a convolutional neural network with a standard laptop. The solution turned to cloud computing. The choice fells on Google Cloud Platform because of the availability of free tier that consists in 300$ free credits that can be used in any GCP product.

We have tested our CNN models on a custom instance of Compute Engine. Our VMs configuration is presented in Table 2

Operating System	CPU	Memory	Disk	GPU	Availability zone
Ubuntu 18.04 LTS	8 core	52 GB	SSD / 100 GB	1x NVIDIA Tesla K80	europe-west1-b

Table 2: *Virtual machine configuration*

Our models have been implemented in keras using tensorflow as a backend. The code is available on our GitHub repository: https://github.com/albertobezzon/cognitiveservices

6 Conclusion

Bibliography

[1] Roberto A. Novoa Justin Ko Susan M. Swetter Helen M. Blau & Sebastian Thrun Andre Esteva Brett Kuprel. "Dermatologist-level classification of skin cancer with deep neural networks." In: (2017) (cit. on p. 1).

[2] Somayeh Karimijeshni Amirreza Rezvantalab Habib Safigholi. "Dermatologist Level Dermoscopy Skin Cancer Classification Using Different Deep Learning Convolutional Neural Networks Algorithms." In: (2018) (cit. on pp. 2, 6).

[3] Abhishek Verma Phillip Ly Doina Bein. "New Compact Deep Learning Model for Skin Cancer Recognition." In: (2018) (cit. on pp. 2, 6).

[4] Cliff Rosendahl & Harald Kittler Philipp Tschandl. "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions." In: (2018) (cit. on p. 2).