

Skin cancer classification using CNNs

Cognitive services course

Alberto Bezzon
1211016

Tommaso Carraro
1210937



UNIVERSITÀ
DEGLI STUDI
DI PADOVA



DIPARTIMENTO
MATEMATICA

Why did we choose skin cancer classification task?

- most widespread cancer
- early detection increases survival rate
- detecting ways are time consuming and have a low precision

Deep learning and CNNs reached good performances in this task and they are able to outperform expert dermatologists

- Transfer learning across all the layers of the Google Inception v3 model pre-trained on ImageNet
 - performances that outperforms expert dermatologists
 - accuracy: 72%
- Compare of common pre-trained models on HAM10000 dataset
 - better results on DenseNet 201
 - Micro - macro F1 scores: 89.01% - 85.13%
- Binary skin cancer classification (benign/malign) on a balanced dataset using a CNN model built from scratch
 - accuracy: 86%

- Difficulties in finding the right dataset
 - bad quality images
 - not biopsy-proven images
- We chose HAM10000 dataset, composed of the most common skin lesions:

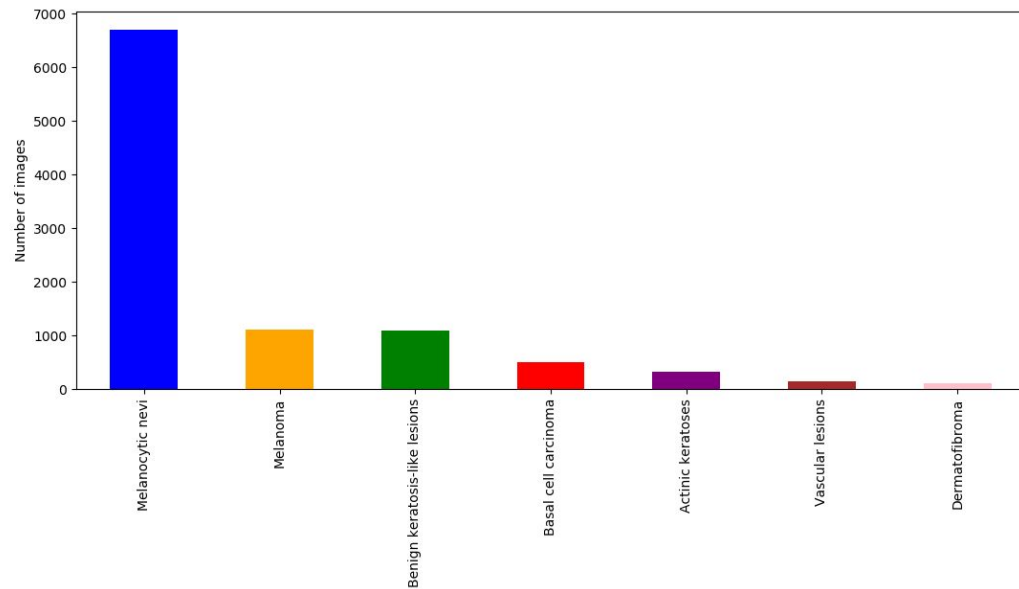
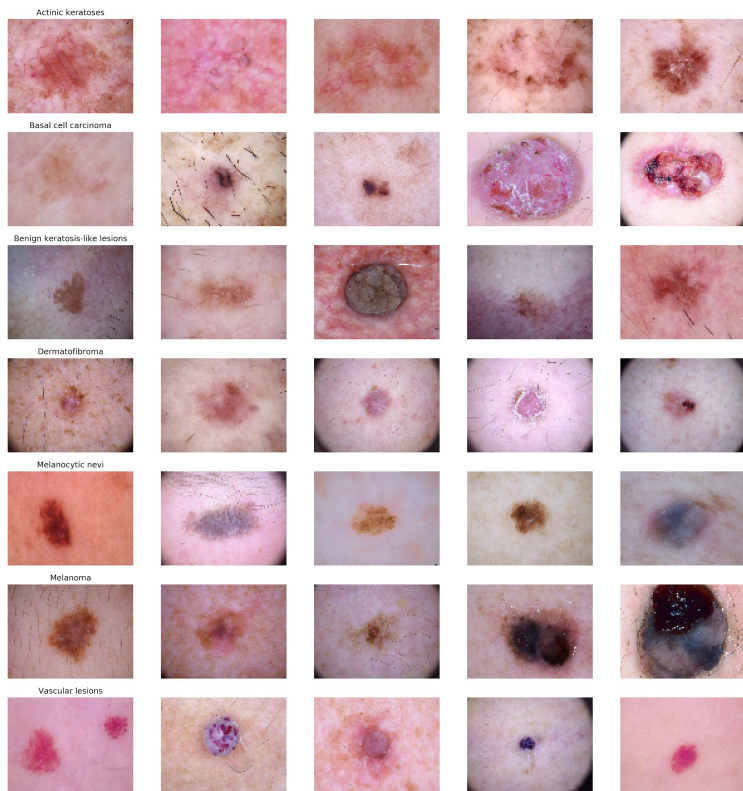
Benign skin lesions:

- Actinic keratosis
- Benign keratosis
- Dermatofibroma
- Melanocytic nevi
- Vascular skin lesions

Malign skin lesions:

- Basal cell carcinoma
- Melanoma

Dataset - skin lesions distribution



Dataset split:

- 80% training set (10% validation set)
- 20% test set

Accuracy

- wrong choice in case of imbalanced data (high accuracy due to bias toward majority classes)

F1-measure

- our goal: obtain a model that maximize both precision and recall on all classes

F1 macro-average

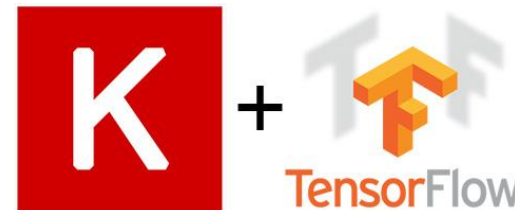
- high macro average values if model performs well on minority classes

Compute engine - Google Cloud Platform



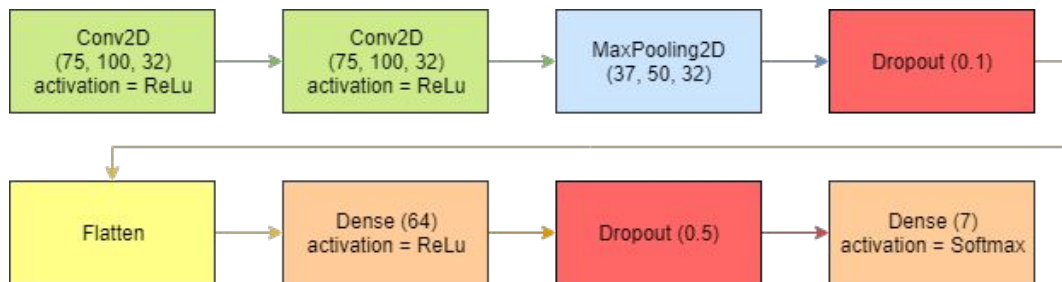
Operating System	CPU	Memory	Disk	GPU
Ubuntu 18.04 LTS	8 core	52 GB	SSD / 100 GB	1x NVIDIA Tesla K80

Software: Keras using Tensorflow as a backend



- Inception v3 model, pre-trained on ImageNet dataset
- classification part on top of it changed with a custom classifier:
 - 1 global average pooling layer
 - 2 fully connected layers
 - softmax layer
- model trained freezing all the layers except of the custom classifier
- 67% accuracy obtained, but all images on the test set were classified like a Melanocytic nevi (majority class)

- simple model built from scratch and composed of:
 - convolutional blocks
 - 2 x convolutional layers 3x3 filter size
 - max pooling layer
 - classification part
- we tried different architectures beginning from a base model and trying to improve it adding convolutional blocks



Performances of tried models



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 $2xConv2D(32) \rightarrow MaxPool2D \rightarrow Dropout(0.1) \rightarrow Flatten \rightarrow FC(64) \rightarrow Dropout(0.5) \rightarrow Softmax(7)$

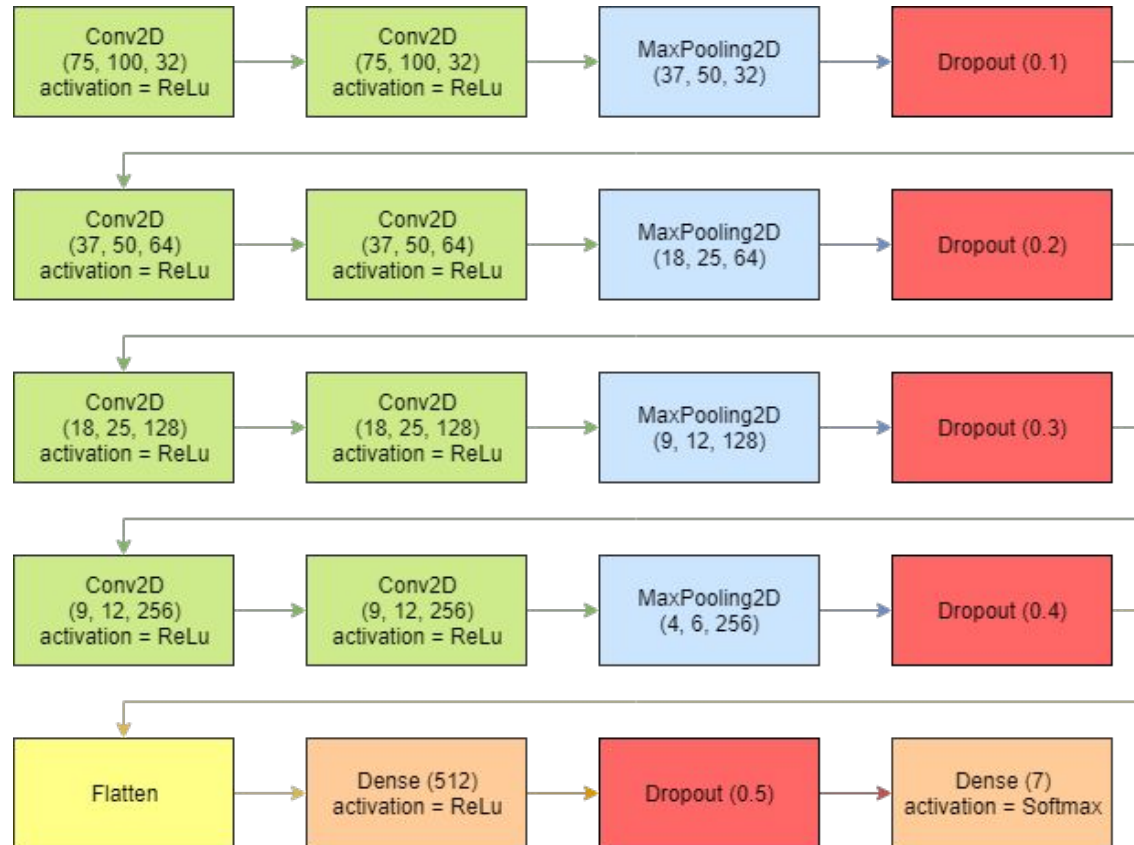
² $2xConv2D(32) \rightarrow MaxPool2D \rightarrow Dropout(0.1) \rightarrow 2xConv2D(64) \rightarrow MaxPool2D \rightarrow Dropout(0.2) \rightarrow Flatten \rightarrow FC(128) \rightarrow Dropout(0.5) \rightarrow Softmax(7)$

³ $2xConv2D(32) \rightarrow MaxPool2D \rightarrow Dropout(0.1) \rightarrow 2xConv2D(64) \rightarrow MaxPool2D \rightarrow Dropout(0.2) \rightarrow 2xConv2D(128) \rightarrow MaxPool2D \rightarrow Dropout(0.3) \rightarrow Flatten \rightarrow FC(256) \rightarrow Dropout(0.5) \rightarrow Softmax(7)$

⁴ $2xConv2D(32) \rightarrow MaxPool2D \rightarrow Dropout(0.1) \rightarrow 2xConv2D(64) \rightarrow MaxPool2D \rightarrow Dropout(0.2) \rightarrow 2xConv2D(128) \rightarrow MaxPool2D \rightarrow Dropout(0.3) \rightarrow 2xConv2D(256) \rightarrow MaxPool2D \rightarrow Dropout(0.4) \rightarrow Flatten \rightarrow FC(512) \rightarrow Dropout(0.5) \rightarrow Softmax(7)$

⁵ $2xConv2D(32) \rightarrow MaxPool2D \rightarrow Dropout(0.1) \rightarrow 2xConv2D(64) \rightarrow MaxPool2D \rightarrow Dropout(0.2) \rightarrow 2xConv2D(128) \rightarrow MaxPool2D \rightarrow Dropout(0.3) \rightarrow 2xConv2D(256) \rightarrow MaxPool2D \rightarrow Dropout(0.4) \rightarrow 2xConv2D(512) \rightarrow MaxPool2D \rightarrow Dropout(0.5) \rightarrow Flatten \rightarrow FC(1024) \rightarrow Dropout(0.5) \rightarrow Softmax(7)$

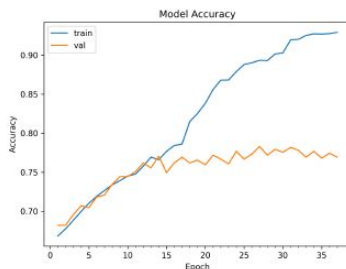
Proposed model architecture



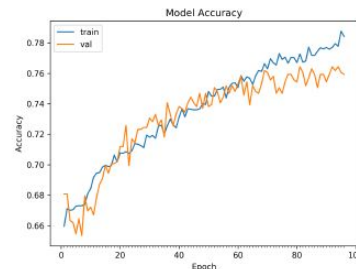
Training settings



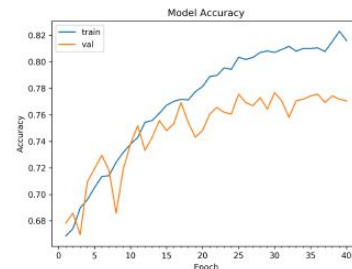
- Adam optimizer with 0.0001 learning rate
- categorical cross entropy loss function
- 32 batch size
- 200 training epochs
- regularization techniques:
 - dropout layers
 - learning rate decay
 - early stopping



(a) *Learning rate 0.001*

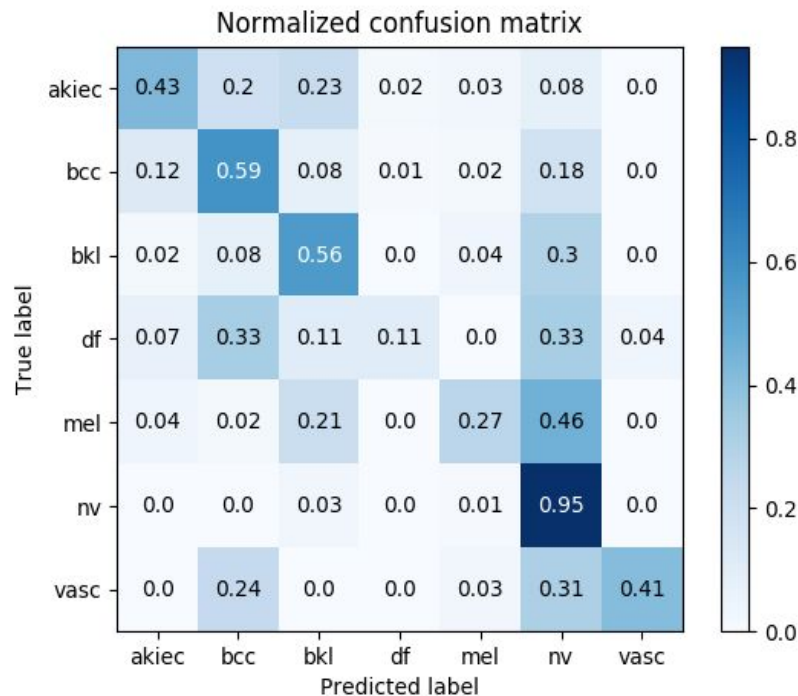


(b) *Learning rate 0.00001*



(c) *Learning rate 0.0001*

Proposed model performances



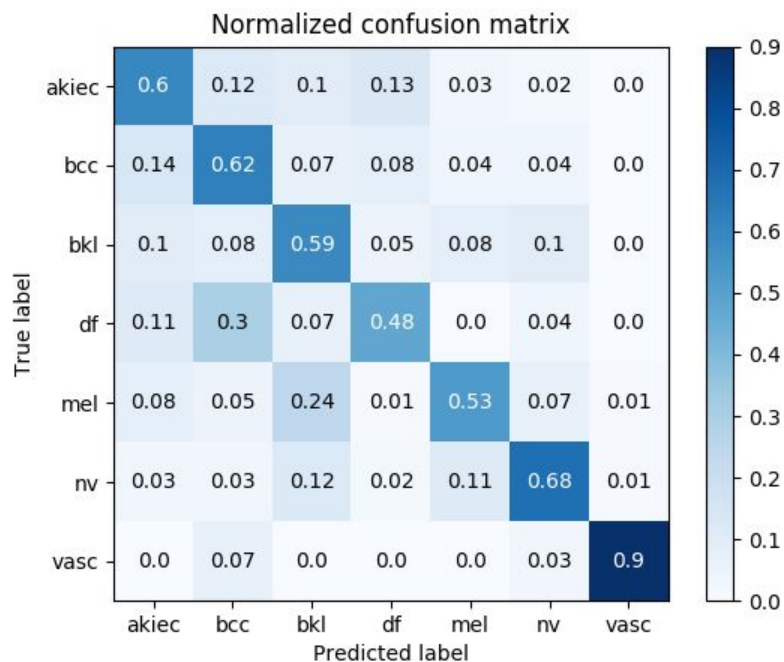
- Difficulties in predicting dermatofibromas and melanomas
- Bias toward the majority class (Melanocytic nevi)
- Different techniques tried to deal with the problem of imbalanced data

We tried two modern techniques to try to improve our proposed model performances:

- Class weighting: try to make the dataset balanced without modify it
- Data oversampling: consists in using data augmentation to make the dataset balanced

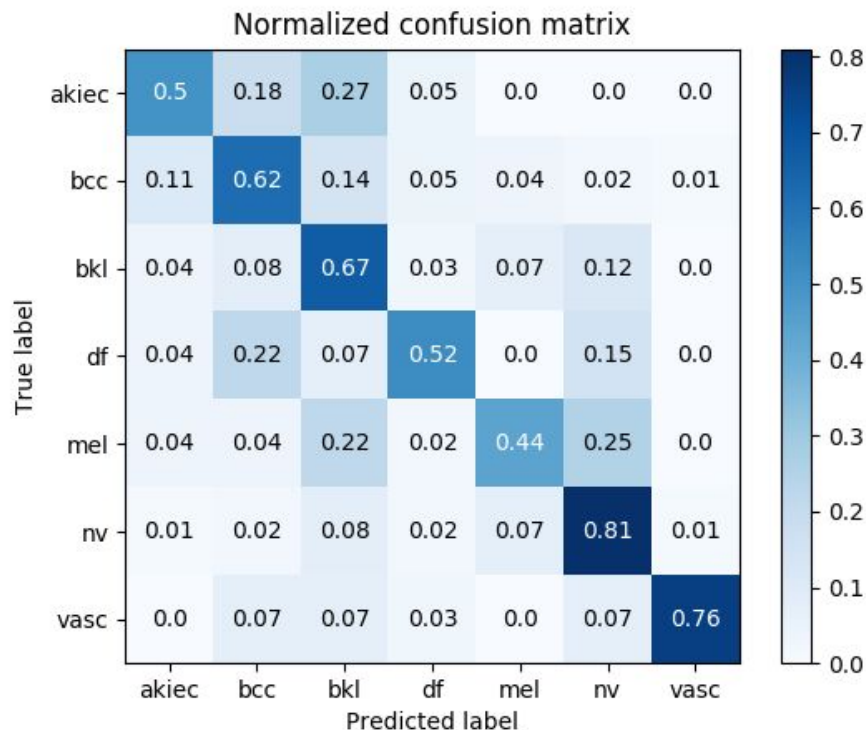


Class weighting performances



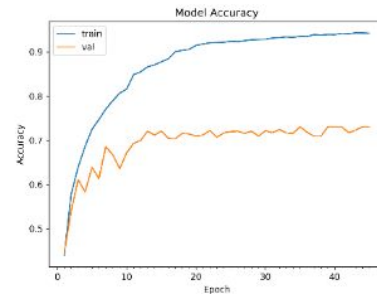
- Loss function penalization: give more emphasis to minority classes and less emphasis to majority classes
- Model performances increased
 - less difficulties in predicting dermatofibromas and melanomas
 - less bias toward the majority class

Oversampling performances



➤ Data augmentation parameters:

- rotation range: 45
- zoom range: 0.2
- width shift range: 0.1
- height shift range: 0.1
- horizontal flip
- vertical flip
- fill mode: nearest



➤ Overfitting due to aggressive data augmentation on minority classes

➤ Reintroduces difficulties in distinguish between classes

- We obtained the best performances on the test set with the class weighting model
 - this shows that more data will help the model to achieve better performances
 - bad performances compared to related works but we tried different approaches respect to transfer learning
- Future works:
 - oversampling with higher quality images
 - try the one vs all approach on HAM10000
 - try transfer learning across all the layers of the pre-trained model
 - test our model on other skin cancer datasets