Deep Learning for skin cancer detection

Binary classification using Convolutional Neural Network

Michele Cernigliaro, ID number 1869097
Sapienza University of Rome
Big Data For Official Statistics

TABLE OF CONTENTS

01

SKIN MELANOMA

Introduction to the medical issue and to the usefulness of deep learning

SKIN MELANOMA

Clinical Issue

- Skin melanoma is one of the deadliest form of skin cancer
- It’s caused by skin cells that begin to develop abnormally
- Melanoma usually detected as:
    - Appearance of a new mole
    - Change in an existing mole

SKIN MELANOMA

When it occurs

- Melanomas typically occur in the skin
  - But it can involve also organs
- They’re more common in:
  - People with pale skin and freckles
  - People heavily exposed to ultraviolet (UV) radiations

SKIN MELANOMA
Why it's dangerous

- It’s a type of skin cancer that can spread to other organs in the body.
- The incidence of contracting a skin cancer is increasing annually
  - Higher increase than any other cancer
- Skin cancer are often diagnosed in people under 50
  - unusually early w.r.t. other types of cancer!

Source: *National Health Service (England):* www.nhs.uk

SKIN MELANOMA

How to detect melanomas

- Skin exams
  - Self examination
  - Examination by a trained professional
- Biopsy

SKIN MELANOMA

Prevention with auto-detection

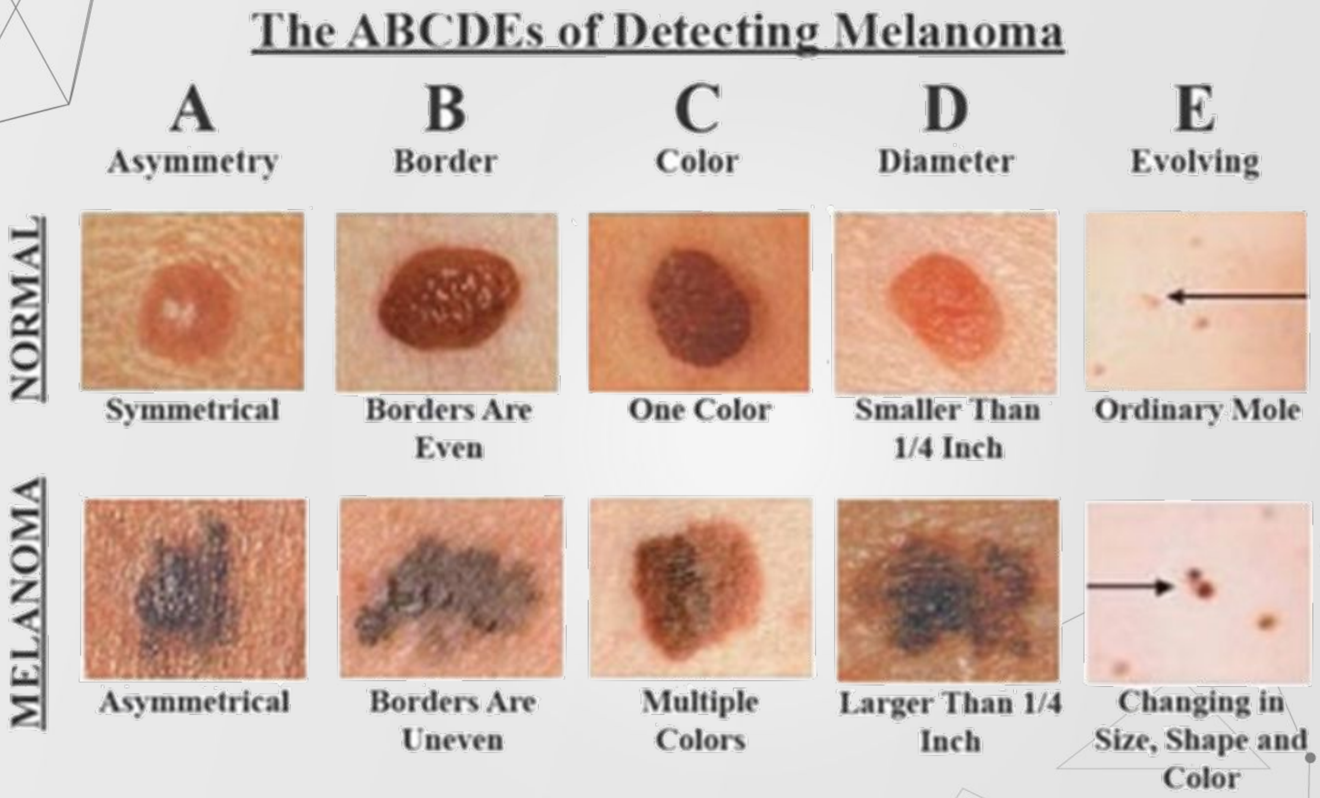


Image courtesy: higherperspective.com

SKIN MELANOMA

Deep Learning for Medical Image Analysis

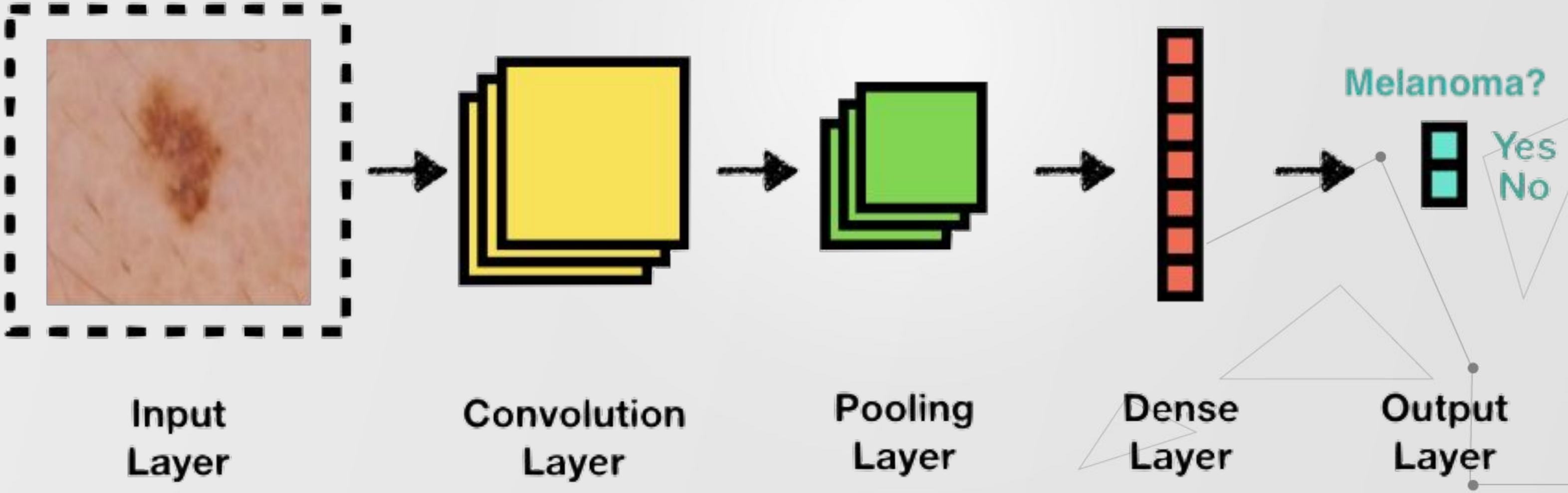
- Deep Learning is the state of the art for Computer vision
- It has a lot of potential in the medicine of the future
    - Huge contribution for disease detection

- A lot of studies in favour of CNNs for Medical Image Analysis
    a. Segmentation
    b. Classification

SKIN MELANOMA

Task of the project

- Model a Convolutional Neural Network
  - Able to recognise from an image if a nevus is a melanoma
- The problem become a **binary classification task!**
- i.e. given an input image:



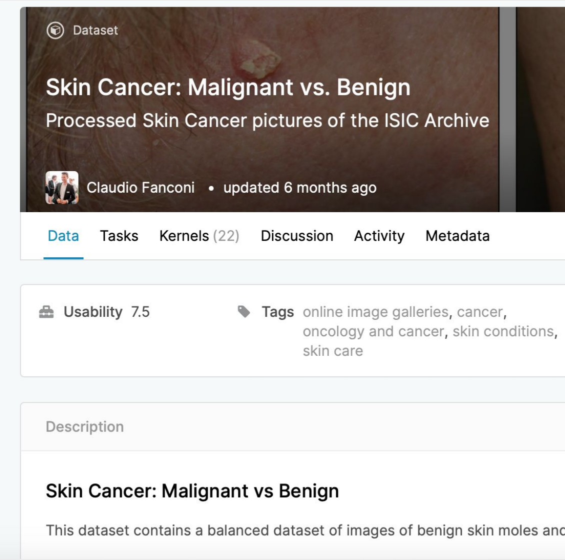
02

DATASET

Dataset description

DATASET
Kaggle Dataset

- We found a dataset of images on Kaggle:
- Processed Skin Cancer picture from **ISIC Archive**
  - 3297 .jpeg images
    - 224x224x3
    - 1800 benign moles images
    - 1497 malignant moles images



https://www.kaggle.com/fanconic/skin-cancer-malignant-vs-benign

DATASET

Kaggle Dataset

Total: 3297 images

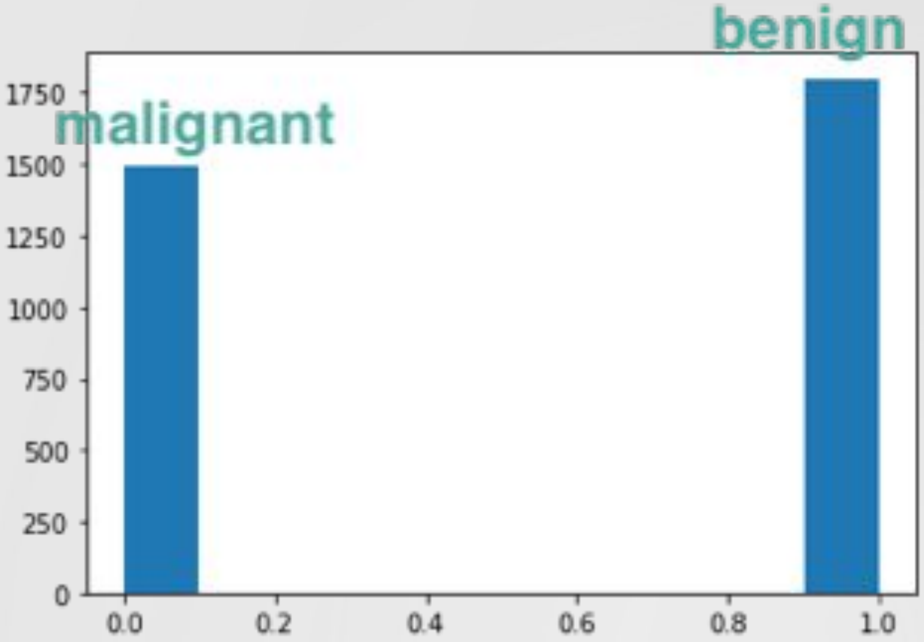
Label **1**: Negative exam   1800 (54.60% of total)

Label **0**: Positive exam   1497 (45.40% of total)

The dataset is a little unbalanced:

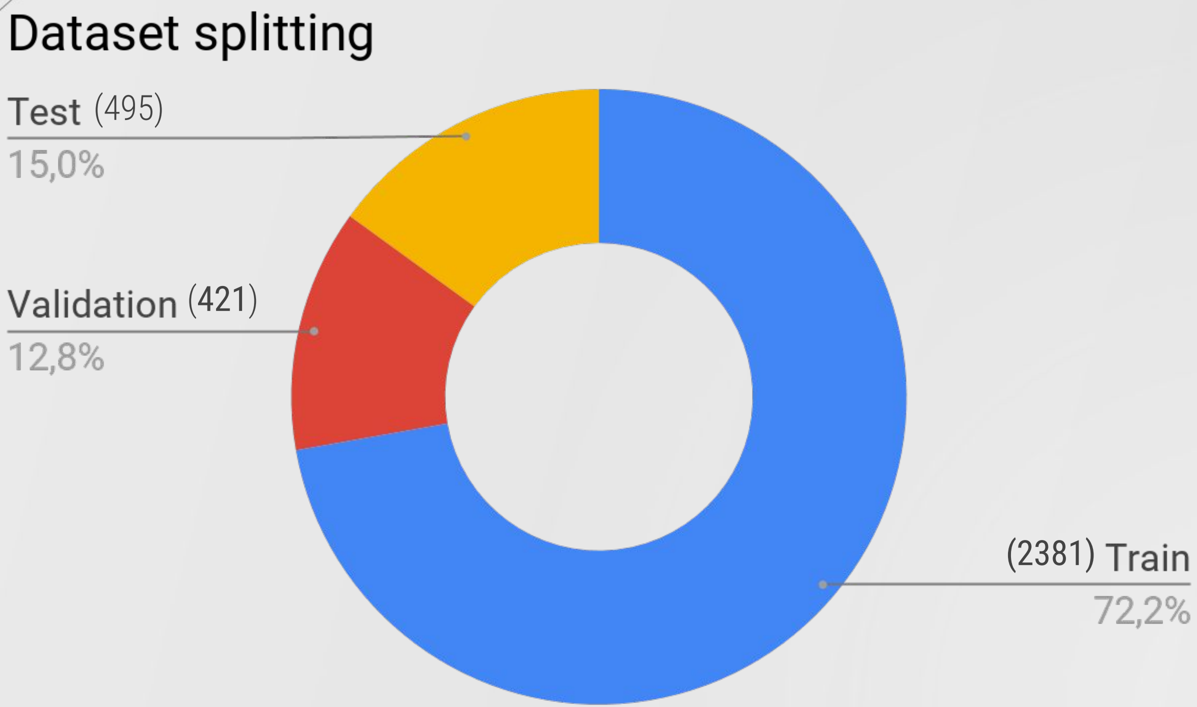
We'll see that this unbalance is **crucial**

in clinical disease detection



DATASET

Train, Validation and Test



03

DATA AUGMENTATION

The necessity of data augmentation

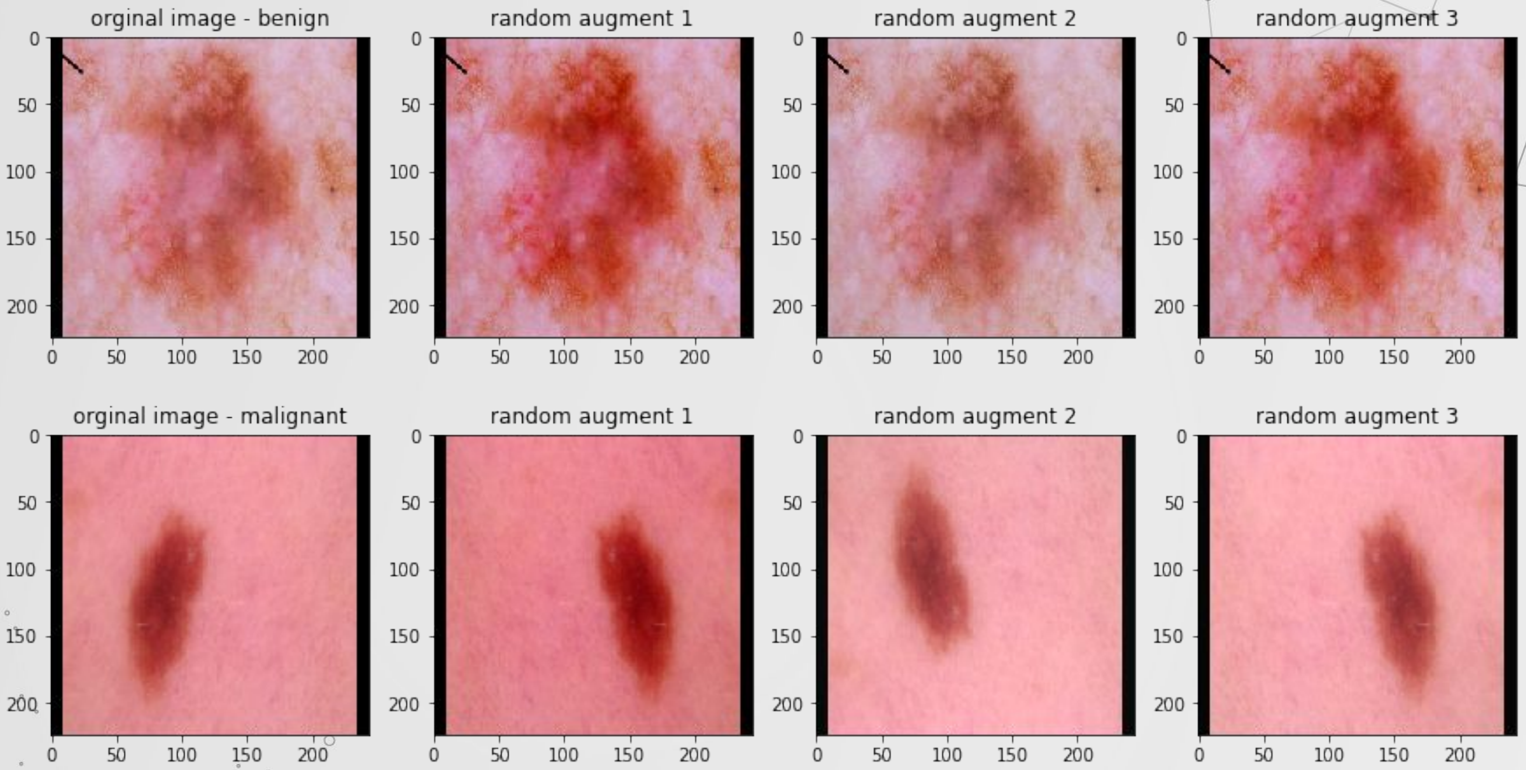
DATA AUGMENTATION

Useful tool for the training

- Data augmentation is a technique to *virtually* increase the training dataset
- While fitting the model the images of the training set are **randomly augmented**
    - Small random changes to the images:
        - Rotation
        - Vertical/horizontal flip
        - Color: saturation, brightness, contrast
- Data augmentation is also useful for overfitting:
    - It makes the network **more robust** to small changes

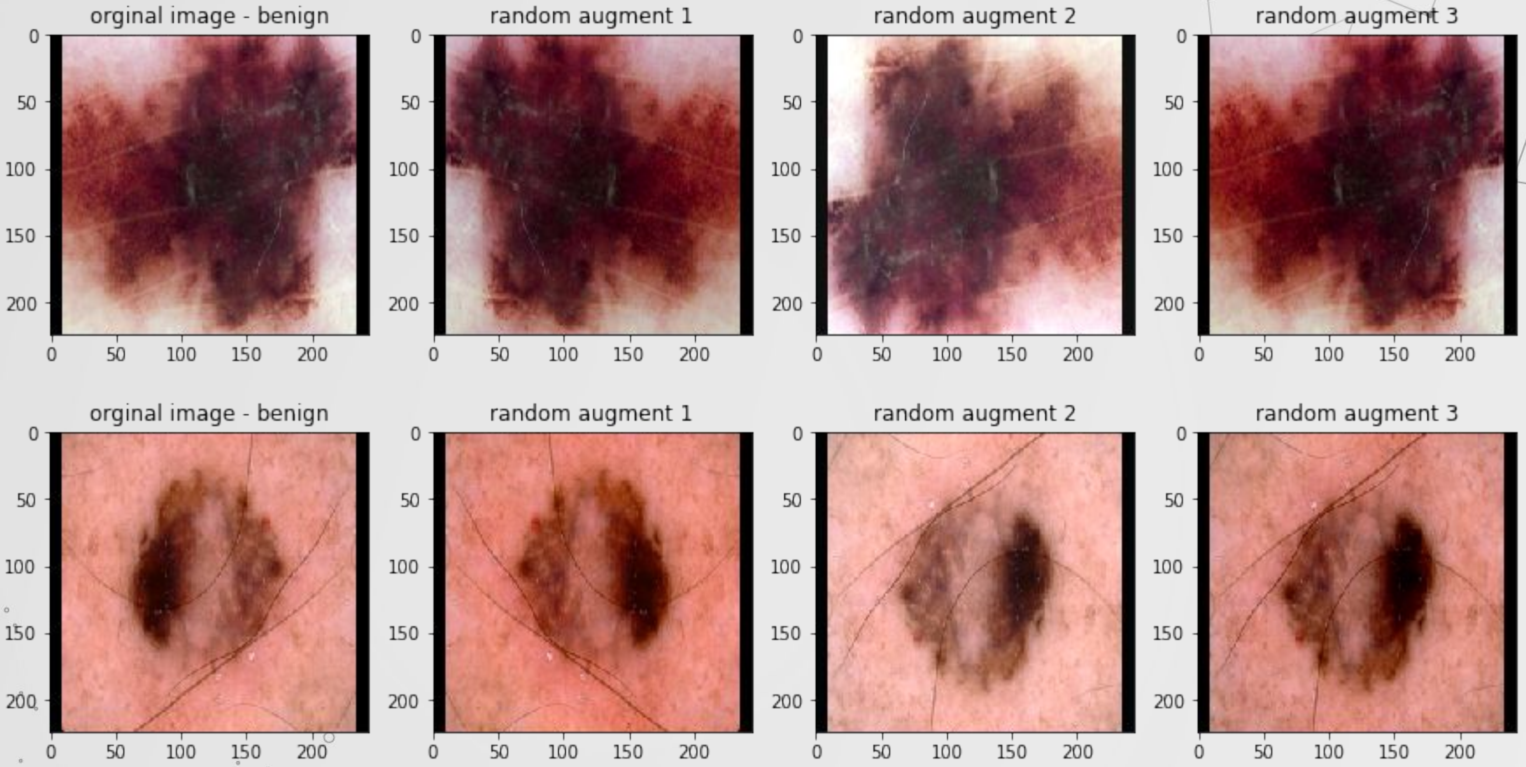
DATA AUGMENTATION

Examples



DATA AUGMENTATION

Examples



04

CNN MODELLING

Elicitation and training of 2 different
Convolutional Neural Networks

CNN MODELLING

Environment

- We used **tensorflow** library and **Keras** API
- We worked on Google Colab free cloud service (GPU)
- Two different approaches

1. Implementing from scratch a CNN
2. Transfer Learning

Importing a pre-trained Neural Network

CNN MODELLING

1. CNN from scratch

- **5 convolutional blocks**
  - 32, 64, 128, 192, 256 filters
  - Zero padding
  - 5x5 filters
  - Custom weights initialization and regularization
  - Batch normalization
  - Relu activation function

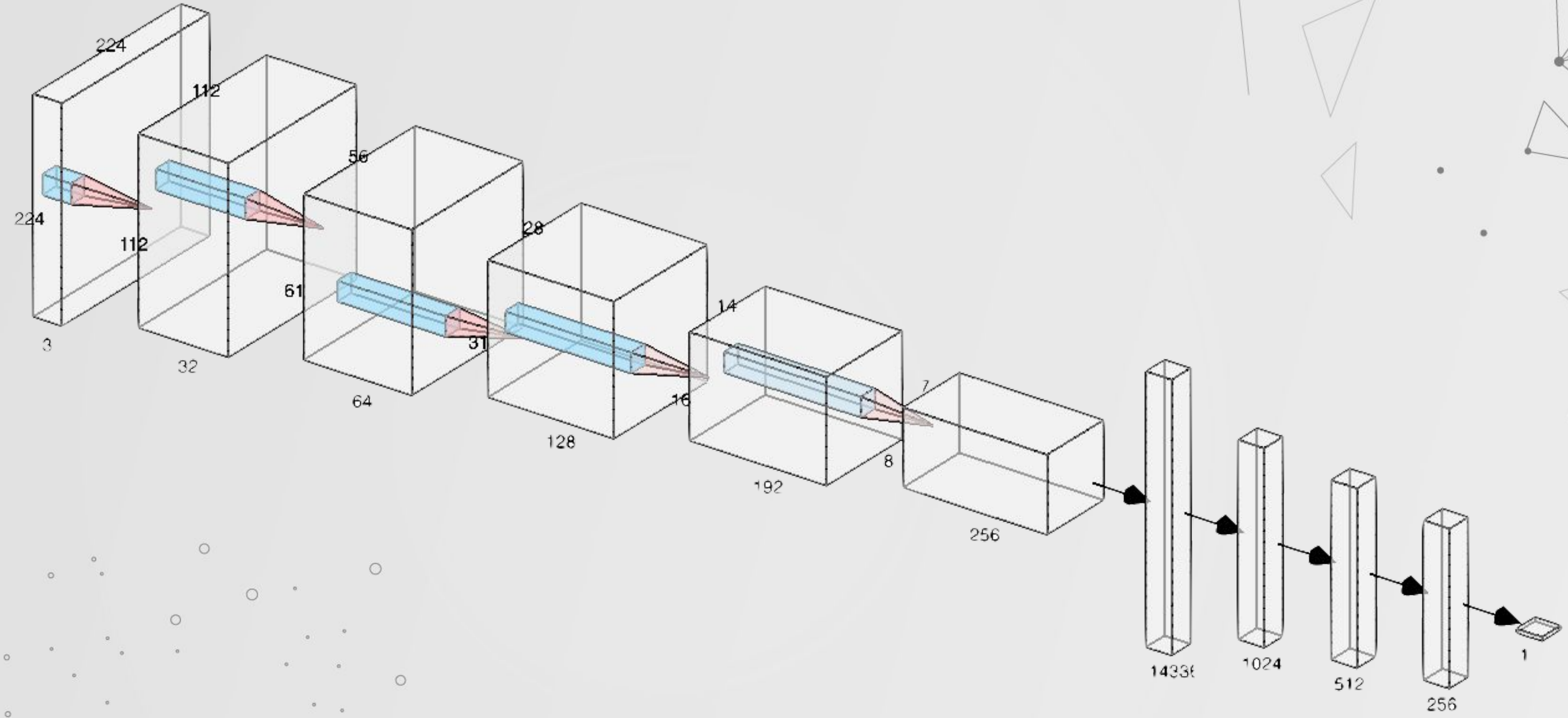
- **Fully connected layer**
  - Ending up with 1 sigmoid function

- **Binary cross-entropy as loss function**

CNN MODELLING

1. CNN from scratch

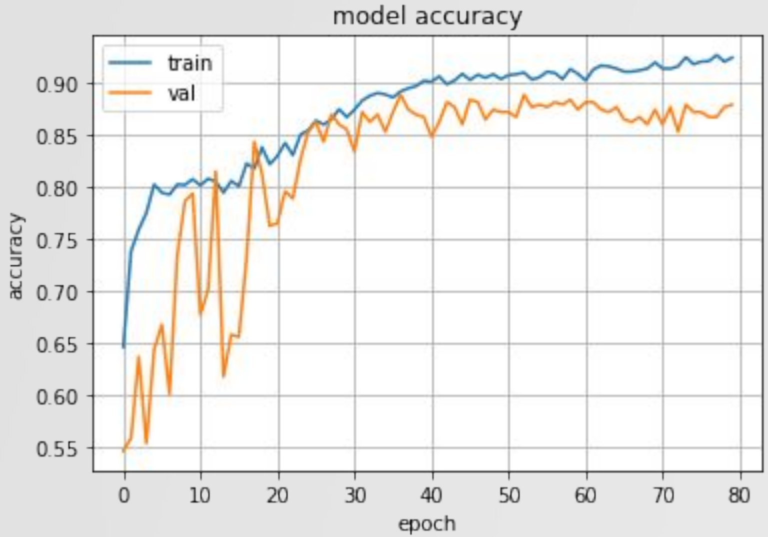
Scheme

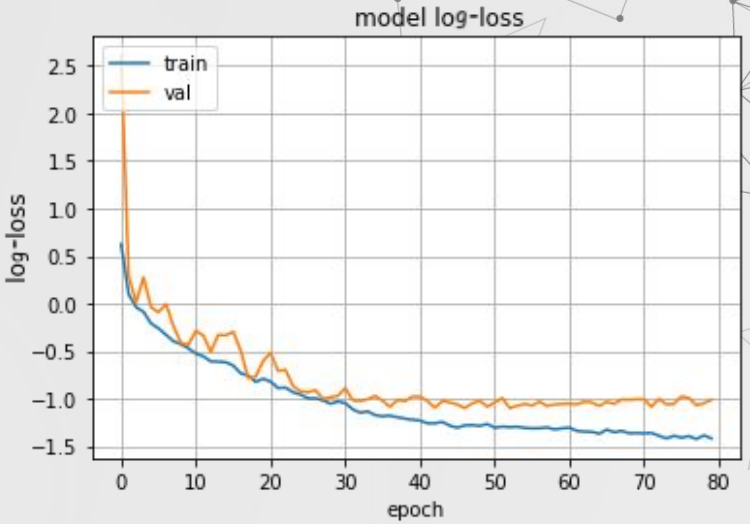


Done with NN-SVG tool by Alex Lenail.

CNN MODELLING

1. CNN from scratch





	Train	Validation	Test
Accuracy	0.912	0.907	0.895
Loss	0.277	0.306	0.319

CNN MODELLING

1. CNN from scratch

Confusion Matrices

	Malignant	Benign
Malignant	941	140
Benign	70	1230

Train

	Malignant	Benign
Malignant	173	18
Benign	21	209

Validation

	Malignant	Benign
Malignant	191	34
Benign	18	252

Test

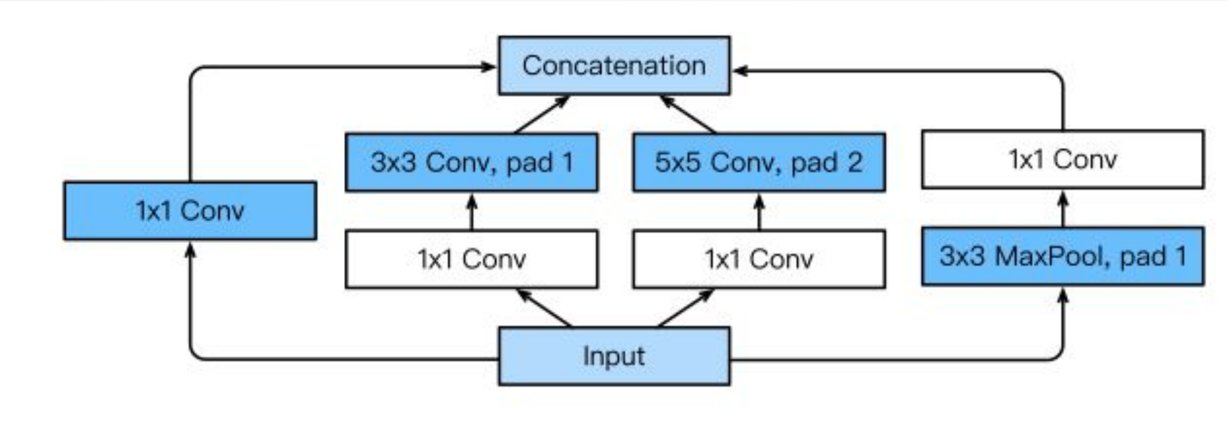
Columns: predictions
Rows: true Values

CNN MODELLING

2. InceptionV3

Transfer Learning

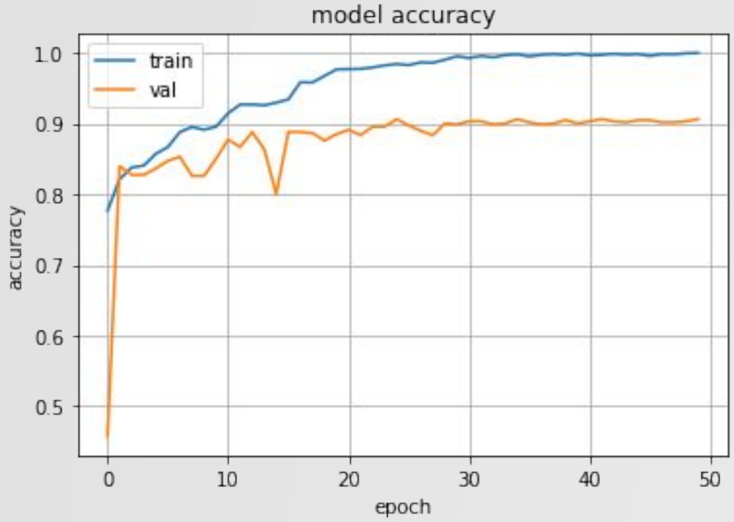
- We took a more complex network already pre-trained
    - **InceptionV3**
    - Weights pre-trained on “imagenet” dataset
- We had to change input and output of the InceptionV3 network
  In order to match our task

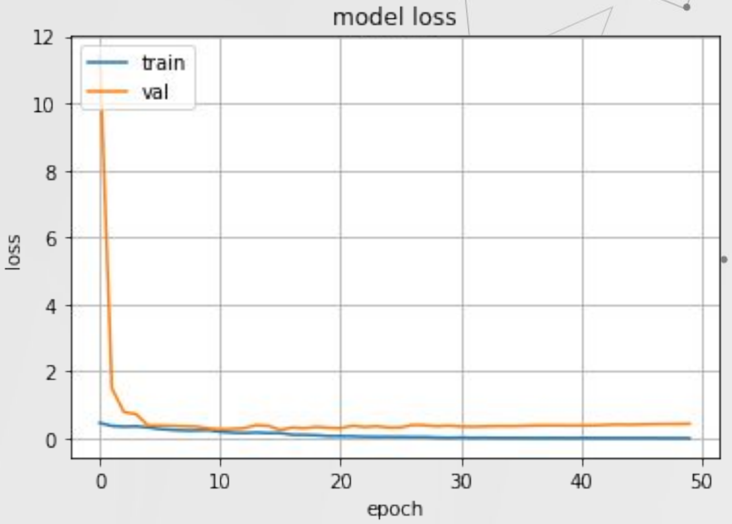


Inception block, source: Dive into Deep Learning, Chapter 7.4.

CNN MODELLING

2. InceptionV3





	Train	Validation	Test
Accuracy	0.948	0.923	0.941
Loss	0.141	0.163	0.159

CNN MODELLING

2. InceptionV3

Confusion Matrices

	Malignant	Benign
Malignant	1018	63
Benign	60	1240

Train

	Malignant	Benign
Malignant	175	16
Benign	17	213

Validation

	Malignant	Benign
Malignant	212	13
Benign	16	254

Test

Columns: predictions
Rows: true Values

05

RESULTS

Comparing the two different models

RESULTS

Model comparisons

- The first model reached a good accuracy
  - Even if it fluctuates too much

- The second model is more stable
  - We clearly see when the model starts overfitting
    - around 16th epoch

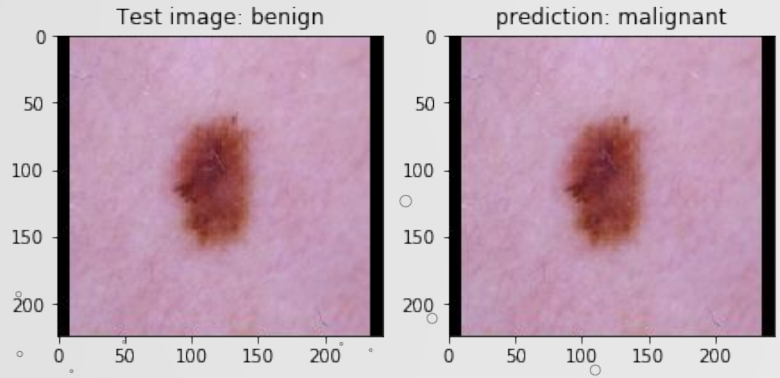
- In both training and testing phases, the second model performs better:
  - In term of accuracy
  - In term of loss

RESULTS

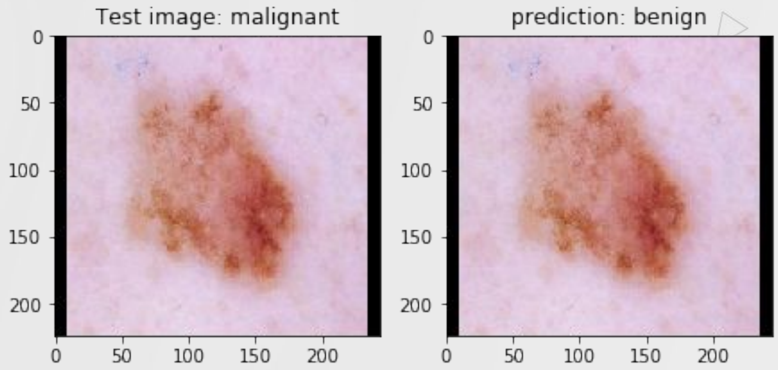
Big issue in clinical diagnosis

Big issue in clinical diagnosis: False Negative

- **The worst mistake**:
    - An exam with negative result (“healthy”) on a positive patient (with melanoma)
    - Predicting as “benign” a melanoma is much worse than predicting as malignant a benign nevus!!!



False positive: still acceptable



False negative: not acceptable

RESULTS

Big issue in clinical diagnosis

Big issue in clinical diagnosis: False Negative

- The first CNN seems to fail (# False Negative > # False Positive)
- The second one has more False Positive than False Negative

	Malignant	Benign
Malignant	191	**34**
Benign	18	252

Confusion matrix (Test set)
First model

	Malignant	Benign
Malignant	212	**13**
Benign	16	254

Confusion matrix (Test set)
Second model

06

Possible improvements

Suggestion on further developments

IMPROVEMENTS

We can surely improve the network

- More data (3000 images are not enough)
- Try different architectures
- Handling False Negative issue
  - pre data augmentation only on melanoma images
- Trying other “paths”
  - i.e. segmented-based classification

IMPROVEMENTS

Beyond binary classification

- Skin melanoma is not the only form of skin cancer:
  - There are different types of pigmented lesions:
    (Actinic keratoses and intraepithelial carcinoma,
    basal cell carcinoma, benign keratosis-like lesions…)

- We can extend the problem to a **multiclass classification**
  - HAM10000 dataset

THANKS

Does anyone have any questions?
cernigliaro.1869097@studenti.uniroma1.it

Project available on Github repository:
https://github.com/cerniello/DeepLearning_SkinCancer

Extra

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 244, 3)]	0
conv2d (Conv2D)	(None, 112, 122, 32)	2432
batch_normalization (BatchNo	(None, 112, 122, 32)	128
activation (Activation)	(None, 112, 122, 32)	0
conv2d_1 (Conv2D)	(None, 56, 61, 64)	51264
batch_normalization_1 (Batch	(None, 56, 61, 64)	256
activation_1 (Activation)	(None, 56, 61, 64)	0
conv2d_2 (Conv2D)	(None, 28, 31, 128)	204928
batch_normalization_2 (Batch	(None, 28, 31, 128)	512
activation_2 (Activation)	(None, 28, 31, 128)	0
conv2d_3 (Conv2D)	(None, 14, 16, 192)	614592
batch_normalization_3 (Batch	(None, 14, 16, 192)	768
activation_3 (Activation)	(None, 14, 16, 192)	0
conv2d_4 (Conv2D)	(None, 7, 8, 256)	1229056
batch_normalization_4 (Batch	(None, 7, 8, 256)	1024
activation_4 (Activation)	(None, 7, 8, 256)	0
flatten (Flatten)	(None, 14336)	0
dense (Dense)	(None, 1028)	14738436
dense_1 (Dense)	(None, 516)	530964
dropout (Dropout)	(None, 516)	0
dense_2 (Dense)	(None, 256)	132352
dropout_1 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 1)	257

Total params: 17,506,969
Trainable params: 17,505,625
Non-trainable params: 1,344

Extra

Layer (type)	Output Shape	Param #
inception_v3 (Model)	(None, 2048)	21802784
flatten_1 (Flatten)	(None, 2048)	0
dense_4 (Dense)	(None, 100)	204900
dropout_2 (Dropout)	(None, 100)	0
dense_5 (Dense)	(None, 1)	101

Total params: 22,007,785
Trainable params: 21,973,353
Non-trainable params: 34,432