COVID-19 SCAN DETECTION USING CNN

Vision and Perception course - Project presentation

SAPIENZA
UNIVERSITÀ DI ROMA

Michele Ciciolla Nicolò Mantovani Flavio Lorenzi

How the presentation is organized

Section 1 - The Chosen Dataset & Background

Section 2 - The PreTrained CNN Network - ResNet101

Section 3 - Our custom CNN Network

How the presentation is organized

Section 1 - The Chosen Dataset & Background

Section 2 - The PreTrained CNN Network - ResNet101

Section 3 - Our custom CNN Network

Section 1 - The Community Effort

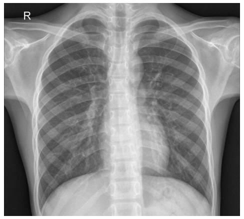
elsevier.com/connect/infographic-global-research-trends-in-infectious-disease

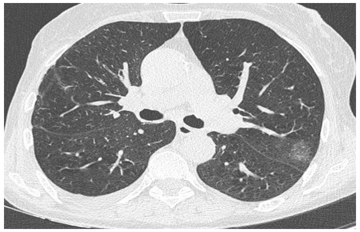
researchers worldwide have been working to advance our understanding of these epidemics and how to respond

2020	78,857
2019	1,658
2018	1,447
2017	1,475

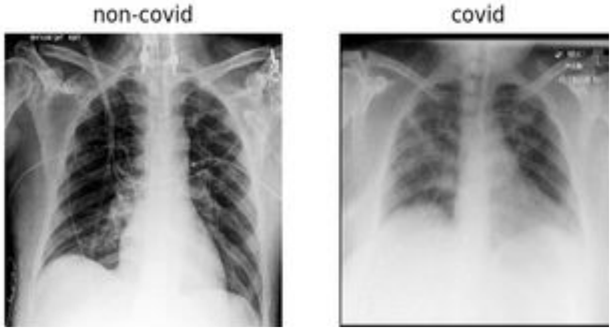
Section 1 - The task

A fast **classification** of the radiological imaging such as chest **X-ray** and chest CT-scan can be helpful to **isolate** the infected persons timely and control this epidemic situation





binary classification

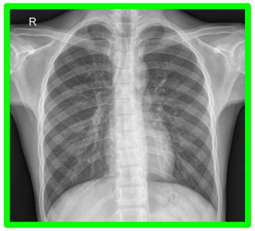


multiclass classification

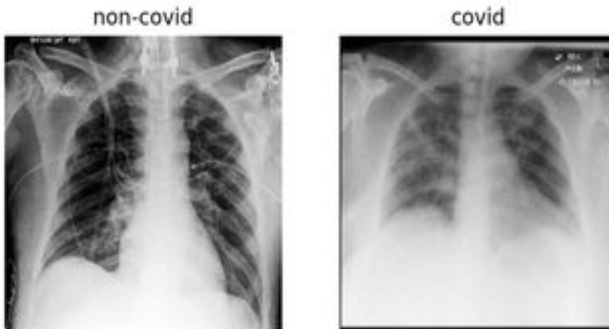
Type	Genus or Species
Viral	COVID-19 (SARSr-CoV-2) SARS (SARSr-CoV-1) Varicella Influenza
Bacterial	*Streptococcus* spp. *Klebsiella* spp. *Escherichia coli* *Mycoplasma* spp. *Legionella* spp. Unknown *Chlamydophila* spp.
Fungal	*Pneumocystis* spp.
Lipoid	Non applicable

Section 1 - The task

A fast **classification** of the radiological imaging such as chest **X-ray** and chest CT-scan can be helpful to **isolate** the infected persons timely and control this epidemic situation



binary classification



multiclass classification

Type	Genus or Species
Viral	COVID-19 (SARSr-CoV-2)
	SARS (SARSr-CoV-1)
	Varicella
	Influenza
Bacterial	*Streptococcus* spp.
	Klebsiella spp.
	Escherichia coli
	Mycoplasma spp.
	Legionella spp.
	Unknown
	Chlamydophila spp.
Fungal	*Pneumocystis* spp.
Lipoid	Non applicable

Section 1 - The task

Table 4

Comparison of the proposed COVID-19 diagnostic method with other deep learning methods developed using radiology images.

Study	Type of Images	Number of Cases	Method Used	Accuracy (%)
Ioannis et al. [43]	Chest X-ray	224 COVID-19(+) 700 Pneumonia 504 Healthy	VGG-19	93.48
Wang and Wong [42]	Chest X-ray	53 COVID-19(+) 5526 COVID-19 (−) 8066 Healthy	COVID-Net	92.4
Sethy and Behra [45]	Chest X-ray	25 COVID-19(+) 25 COVID-19 (−)	ResNet50+ SVM	95.38
Hemdan et al. [41]	Chest X-ray	25 COVID-19(+) 25 Normal	COVIDX-Net	90.0
Narin et al. [44]	Chest X-ray	50 COVID-19(+) 50 COVID-19 (−)	Deep CNN ResNet-50	98
Ying et al. [46]	Chest CT	777 COVID-19(+) 708 Healthy	DRE-Net	86
Wang et al. [47]	Chest CT	195 COVID-19(+) 258 COVID-19(−)	M-Inception	82.9
Zheng et al. [48]	Chest CT	313 COVID-19(+) 229 COVID-19(−)	UNet+3D Deep Network	90.8
Xu et al. [49]	Chest CT	219 COVID-19(+) 224 Viral pneumonia 175 Healthy	ResNet + Location Attention	86.7
Proposed Study	Chest X-ray	125 COVID-19(+) 500 No-Findings	DarkCovidNet	98.08
		125 COVID-19(+) 500 Pneumonia 500 No-Findings		87.02

Section 1 - The Chosen Dataset (1)

COVID-19 Image Data Collection

Joseph Paul Cohen [1 2] Paul Morrison [3] Lan Dao [4]

Abstract

This paper describes the initial COVID-19 open image data collection. It was created by assembling medical images from websites and publications and currently contains 123 frontal view X-rays.

1. Motivation

In the context of a COVID-19 pandemic, is it crucial to streamline diagnosis. Data is the first step to developing any diagnostic tool or treatment. While there exist large public datasets of more typical chest X-rays (Wang et al., 2017; Bustos et al., 2019; Irvin et al., 2019; Johnson et al., 2019; Demner-Fushman et al., 2016), there is no collection of COVID-19 chest X-rays or CT scans designed to be used for computational analysis.

In this paper, we describe the public database of pneumonia cases with chest X-ray or CT images, specifically COVID-19 cases as well as MERS, SARS, and ARDS. Data will be collected from public sources in order not to infringe patient confidentiality. Example images shown in Figure 1.

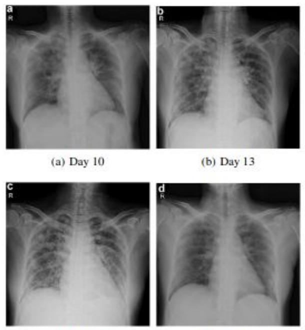
Our team believes that this database can dramatically improve identification of COVID-19. Notably, this would provide essential data to train and test a Deep Learning-based system, likely using some form of transfer learning. These tools could be developed to identify COVID-19 characteristics as compared to other types of pneumonia or in order to predict survival.

Currently, all images and data are released under the following URL: `https://github.com/ieee8023/covid-chestxray-dataset`. As stated above, images collected have already been made public.

2. Expected outcome

This dataset can be used to study the progress of COVID-19 and how its radiological findings vary from other types of

[1]Mila, Quebec Artificial Intelligence Institute [2]University of Montreal [3]Department of Mathematics and Computer Science, Fontbonne University [4]Faculty of Medicine, University of Montreal. Correspondence to: Joseph Paul Cohen <joseph@josephpcohen.com>.



(a) Day 10 (b) Day 13 (c) Day 17 (d) Day 25

Figure 1. Example images from the same patient (#19) extracted from Cheng et al. (2020). This 55 year old female survived a COVID-19 infection.

pneumonia. Similarly to the outcome of the Chest Xray14 (Wang et al., 2017) dataset which enabled significant advances in medical imaging, tools can be developed to predict not only the type of pneumonia, but also its outcome. Eventually, our model could take inspiration from work by Rajpurkar et al. (2017), which could predict pneumonia, as well as Cohen et al. (2019), which deployed such models.

Tools could be built to triage cases in the absence of physical tests, particularly in the context of polymerase chain reaction (PCR) tests shortage (Satyanarayana, 2020; Kelly Geraldine Malone, 2020). These tools could predict patient outcomes such as survival, allowing a physician to plan ahead for specific patients and facilitate management. In extreme situations, where physicians could be faced with the extraordinary decision to choose which patient should be allocated healthcare resources (Yascha Mounk, 2020), such a tool could potentially serve as a measuring device.

We collected scan images found attached to this research paper

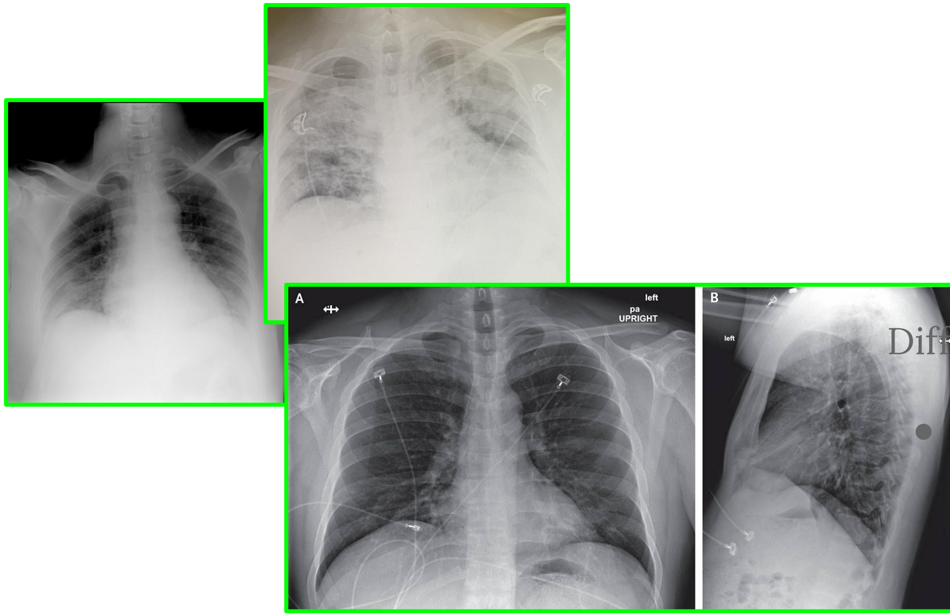
Different resolutions from (255, 249) up to (4280, 3520)

- **COVID-Xray-5k dataset**
  a. with 184 samples containing clinical findings of COVID-19 from 216 patients

Table 1. Number of images per category in COVID-Xray-5k dataset.

Split	COVID-19	Non-COVID
Training Set	84 (420 after augmentation)	2000
Test Set	100	3000

Section 1 - The Chosen Dataset (2)



Different resolutions from (255, 249) up to (4280, 3520)

- **COVID-Xray-5k dataset**
  a. with 84 CT images containing clinical findings of COVID-19 from 216 patients

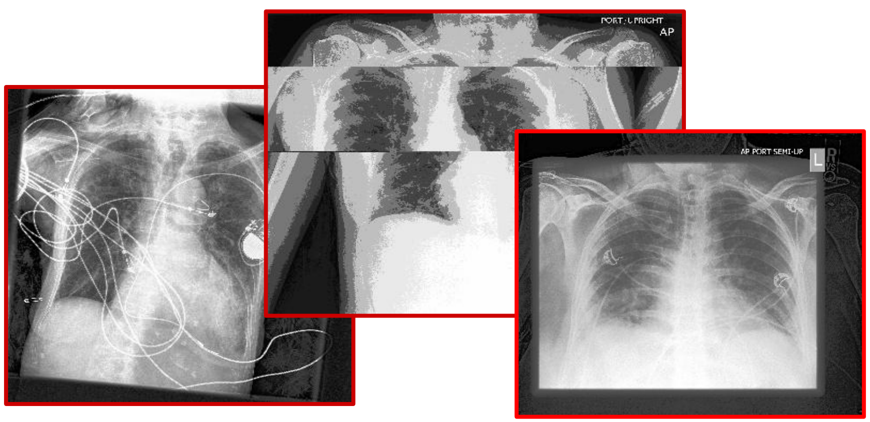
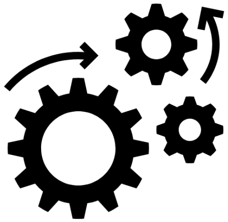


Table 1. Number of images per category in COVID-Xray-5k dataset.

Split	COVID-19	Non-COVID
Training Set	84 (420 after augmentation)	2000
Test Set	100	3000

Section 1 - The Chosen Dataset (2)



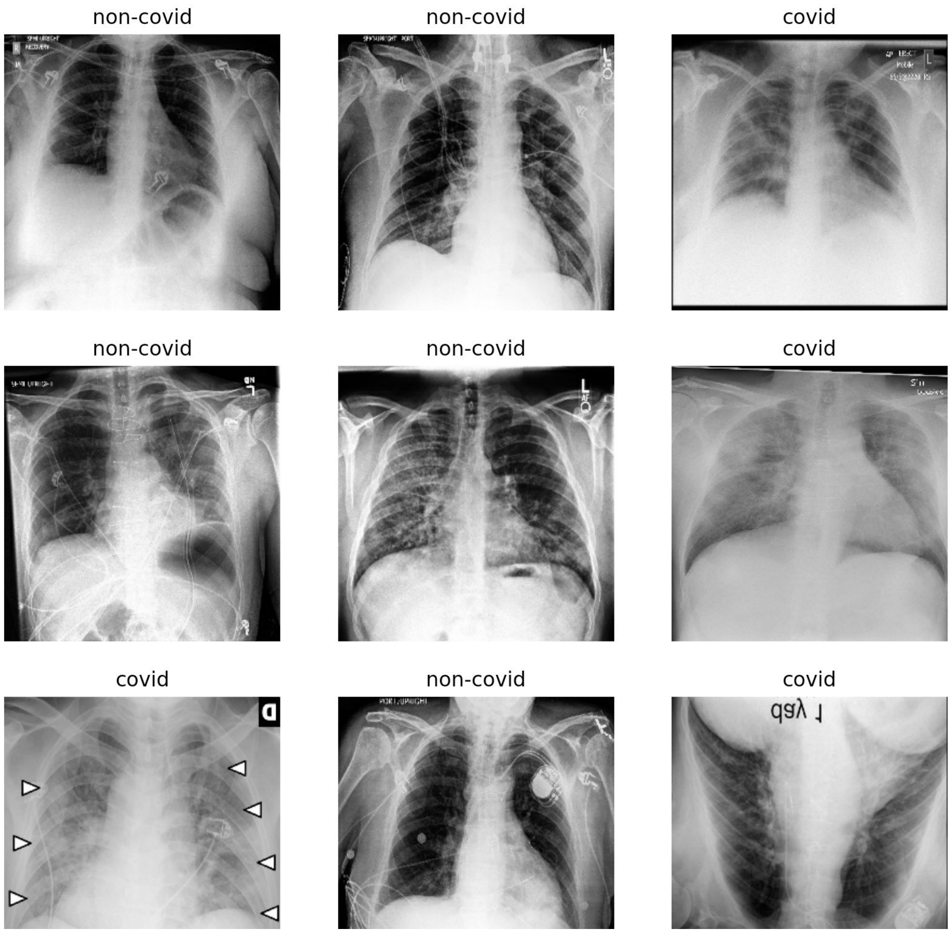
We obtained **more than 1000** images from the initial dataset after the data augmentation phase

Augmentor

1. **rotation** = (few degrees) size preserving rotation
2. **flip_left_right** = mirroring
3. **centered crop**
4. **contrast jitter** = contrast alteration

A. translation
B. gaussian noise
C. vertical mirroring
D. skew

Section 1 - The Chosen Dataset ()



pros:

- community effort
- sharing is caring

cons:

- few images
- overfitting
- need doctors supervision

How the presentation is organized

Section 1 - The Chosen Dataset & Background

Section 2 - The PreTrained CNN Network - ResNet101

Section 3 - Our custom CNN Network

Section 2 - The PreTrained CNN Network - ResNet101 (1)

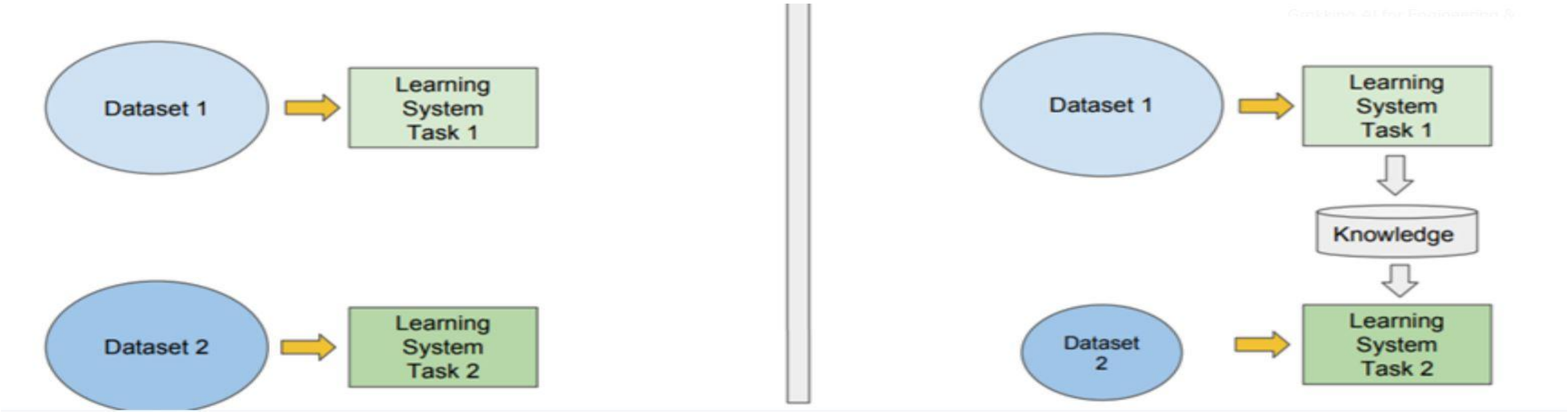
Traditional Machine Learning

- Isolated
- Single Task Learning
- Knowledge not accumulated

vs

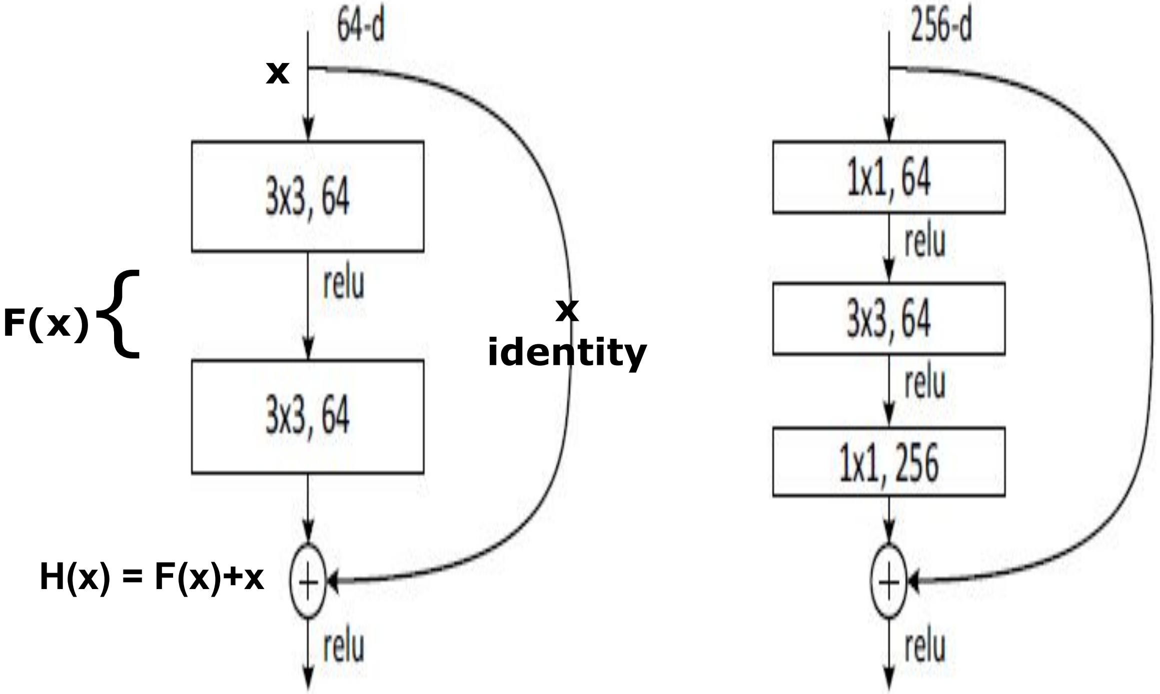
Transfer Learning

- Learning relies on previously learned tasks
- Learning process faster
- More accurate
- Less train data



Section 2 - The PreTrained CNN Network - ResNet101 (1)

- ***"Depth Revolution"***
- ***Vanishing/exploding gradient***
- ***Residual Network:***
  1. Identity Mapping
  2. Projections
- ***Bottleneck design***



Section 2 - The PreTrained CNN Network - ResNet101 (2)

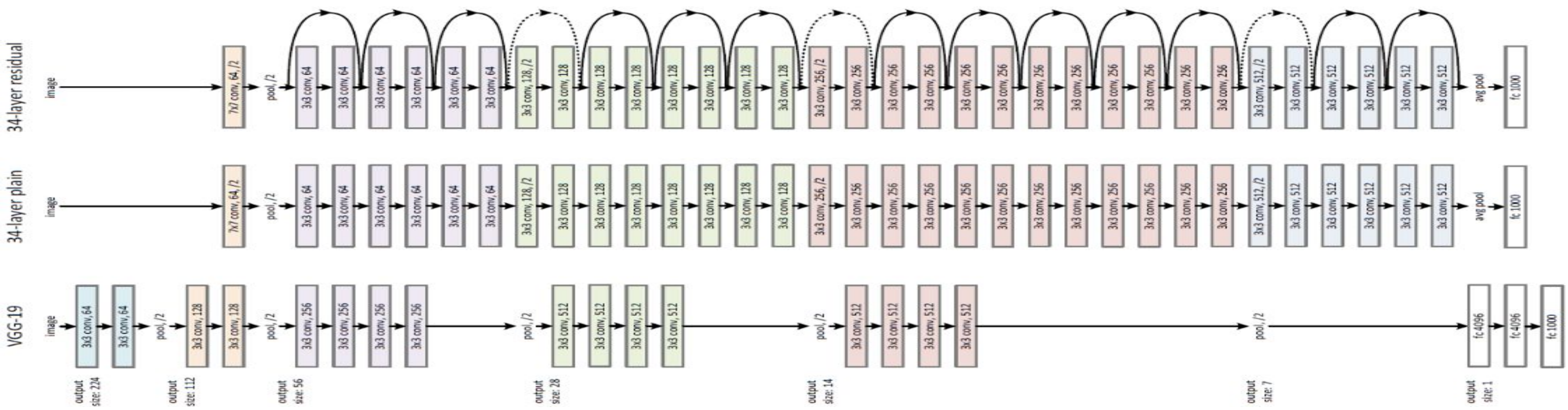
Hyperparameters:

- *Epochs* : 50
- *Batch_size* : 64
- *Optimizer* : SGD
- *Learning rate* : 10^-3
- *Loss* : CrossEntropy
- *Momentum:* 0.9

ImageNet Challenge

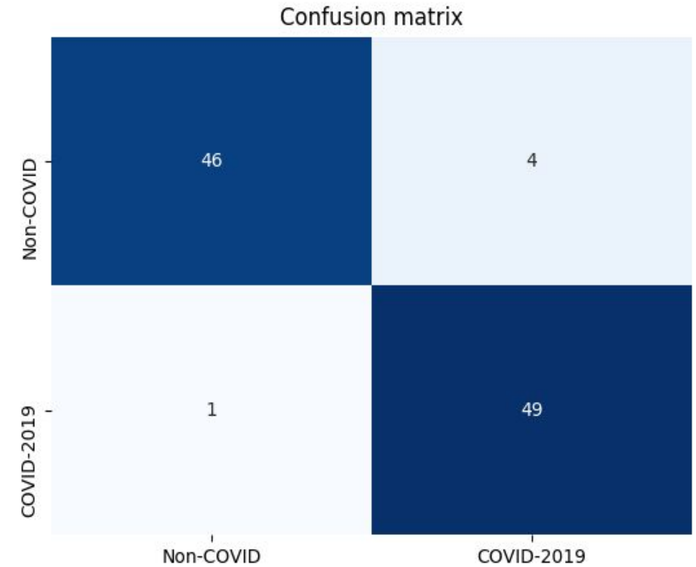
IMAGENET

- 1,000 object classes (categories).
- Images:
  - 1.2 M train
  - 100k test.



Section 2 - The PreTrained CNN Network - ResNet101 (3)

	MIRRORING	MIRRORING ROTATION	MIRRORING ROTATION CONTRAST JITTER	MIRRORING ROTATION CONTRAST JITTER CROP
SENSITIVITY (TP/P)	0.85	0.905	0.95	0.98
SPECIFICITY (TN/N)	0.82	0.86	0.89	0.92
ACCURACY (TP+TN)/(TOT)	0.835	0.88	0.92	0.95



sensitivity : 0.98
specificity : 0.92
accuracy : 0.95
f1_score : 0.9514

How the presentation is organized

Section 1 - The Chosen Dataset & Background

Section 2 - The PreTrained CNN Network - ResNet101

Section 3 - Our custom CNN Network

Section 3 - Our custom CNN Network (1)

Major issues due to the small dataset:

- Overfitting
- Vanishing gradient
- Oscillations in both training and testing

Solutions adopted:

- Batch normalization
- L2 weight decay
- Dropout

```
model = tf.keras.Sequential([
    data_augmentation,
    layers.experimental.preprocessing.Rescaling(1./255)

    layers.Conv2D(16, 5, padding='valid', activation='relu'),
    layers.MaxPooling2D( (2,2) ), #stride default
    layers.BatchNormalization(momentum=0.8),

    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D( (2,2) ), #stride default
    layers.BatchNormalization(momentum=0.8),

    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D( (2,2) ), #stride default
    layers.BatchNormalization(momentum=0.8),

    layers.Conv2D(128, 3, padding='same', activation='relu'),
    layers.MaxPooling2D( (2,2) ), #stride default
    layers.BatchNormalization(momentum=0.8),

    layers.Dropout(0.5),
    layers.Flatten(),

    layers.Dense(256, activation='relu') #First FC layers
    layers.Dense(2)                      #second FC. 2 is our #classes
])
```

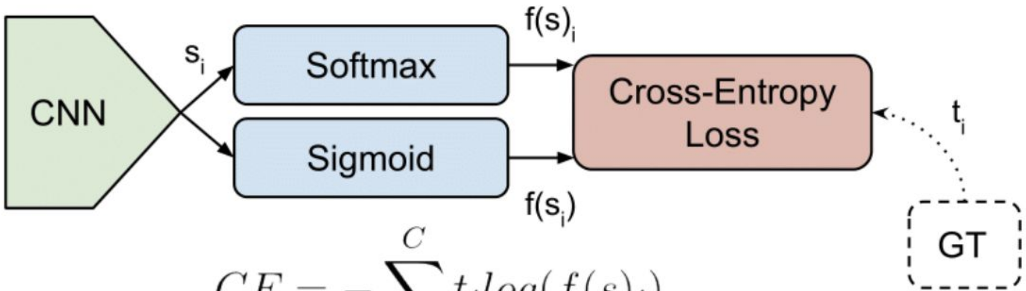
Section 3 - Our custom CNN Network (2)

Loss function experiments:

1. BinaryCrossEntropy
2. CategoricalCrossentropy (Sparse)

The first iterates over two classes (0,1) so it works well with a **sigmoid** before computation.

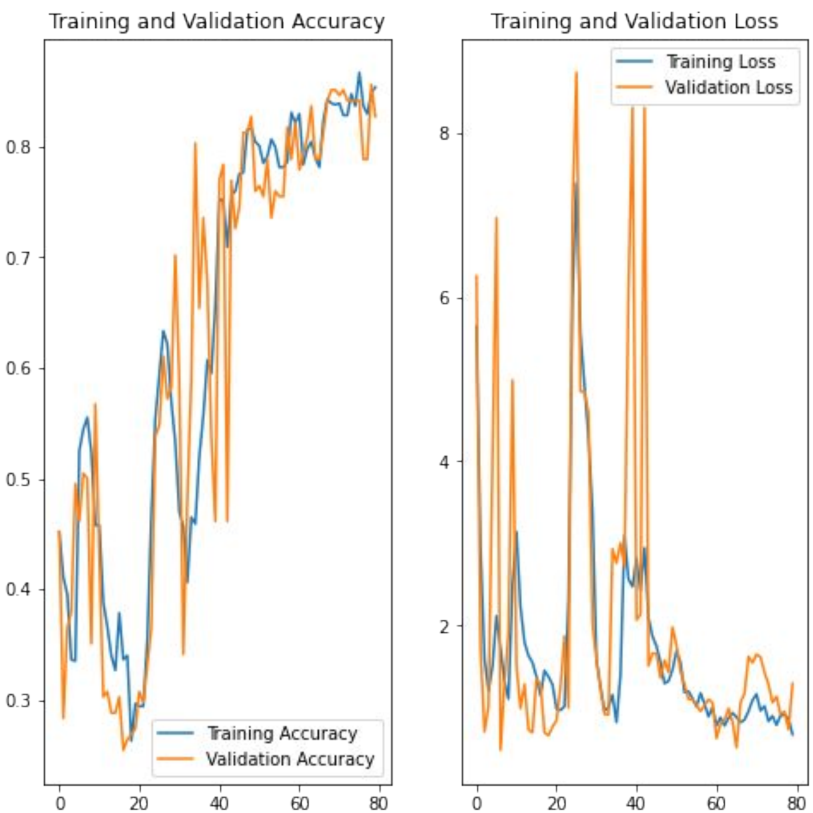
The second is meant for two or more classes, so take advantage of a **softmax** activation.



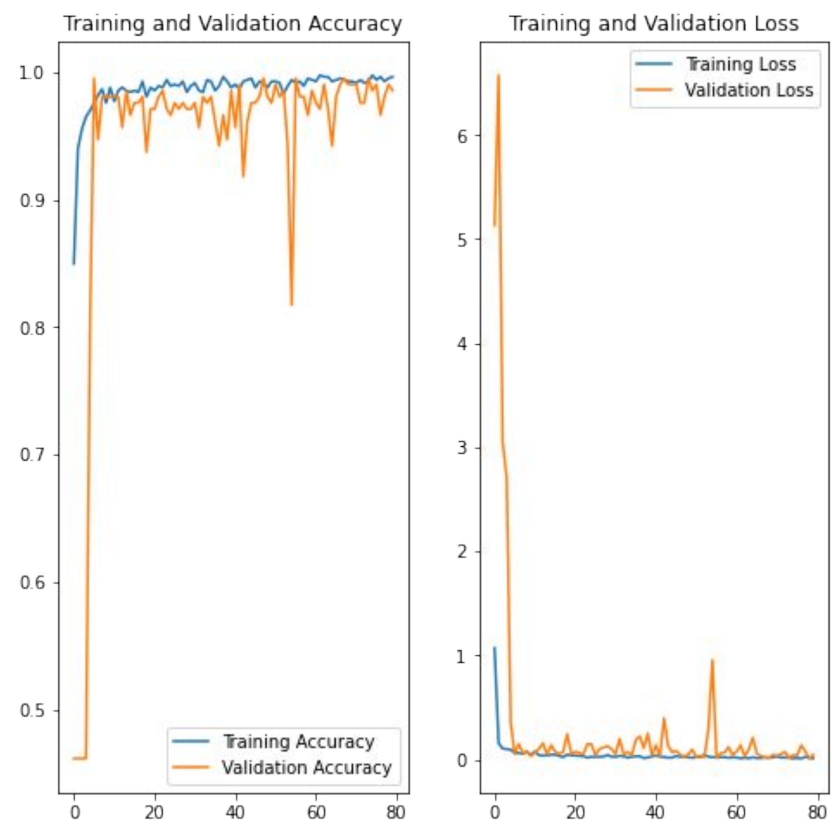
$$CE = -\sum_{i}^{C} t_i log(f(s)_i)$$

$$CE = -\sum_{i=1}^{C'=2} t_i log(f(s_i)) = -t_1 log(f(s_1)) - (1 - t_1) log(1 - f(s_1))$$

Section 3 - Our custom CNN Network (3)



Binary Cross Entropy (best results after a very long hyperparams tuning)



Sparse Cat. Cross Entropy (one of the first obtained results: only 2 epochs to achive an optimal val_acc)

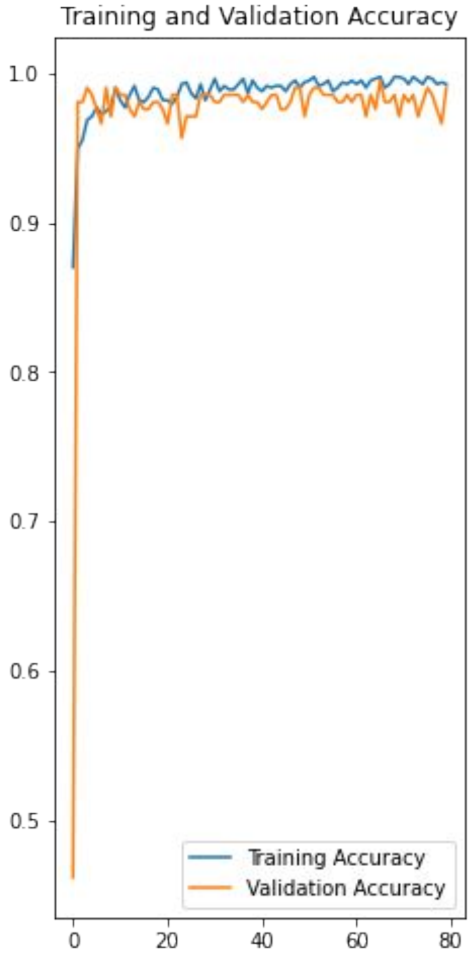
Section 3 - Our custom CNN Network (4)

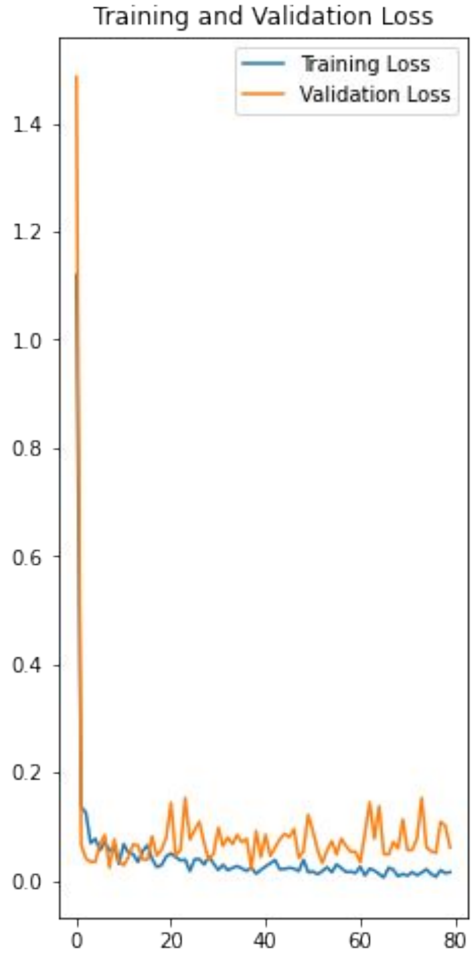
Data augmentation experiments:

	MIRRORING	MIRRORING ROTATION	MIRRORING ROTATION CONTRAST JITTER	MIRRORING ROTATION CONTRAST JITTER CROP
SENSITIVITY (TP/P)	0.71	0.75	0.78	0.86
SPECIFICITY (TN/N)	0.93	0.95	0.96	1.0
ACCURACY (TP+TN)/(TOT)	0.82	0.85	0.87	0.93

Note: some other techniques have been tried, but not relevant in terms of network accuracy

Section 3 - Our custom CNN Network (5)





Best Model

Conv2D: (16) (32) (64) (128)
Dense_1: (256)
Dense_2: (2)

(Not so deep, but very efficient)

Batch size: 64

Epochs: 80

Dropout: 0.5

Batch norm. momentum: 0.8

Loss: sparse categorical cross entropy

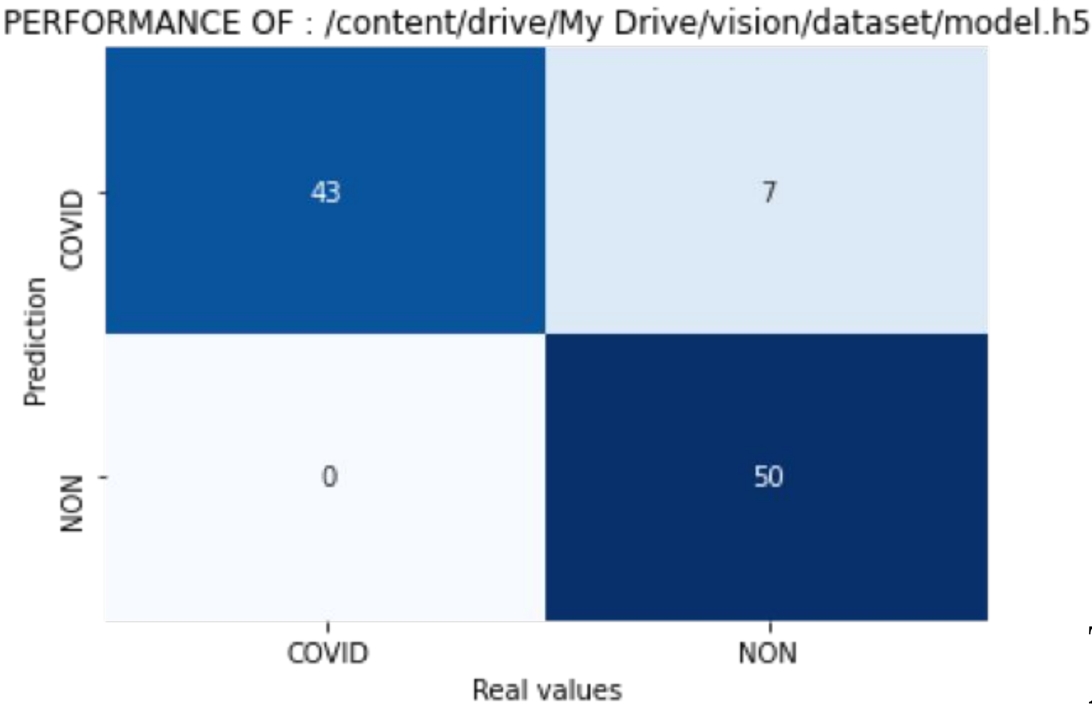
Optimizer: Adam with 0.9 LR decay rate

Starting LR: 1e^-2

Activation function: relu

Section 3 - Our custom CNN Network (6)

To have a practical response of it, a brief **inference** part is realized, with very good results

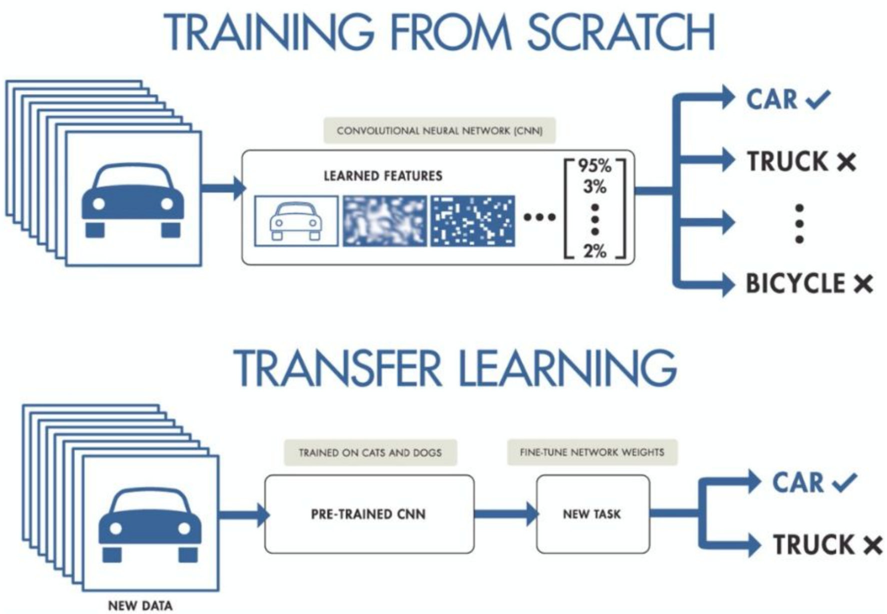


Confusion Matrix

sensitivity : 0.86
specificity : 1.0
accuracy : 0.93
f1_score : 0.9247

The blind testset is the same we used for the first part with the transfer learning, to make also a further comparison

Final considerations

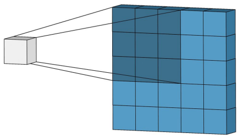


best model accuracy: 0.93

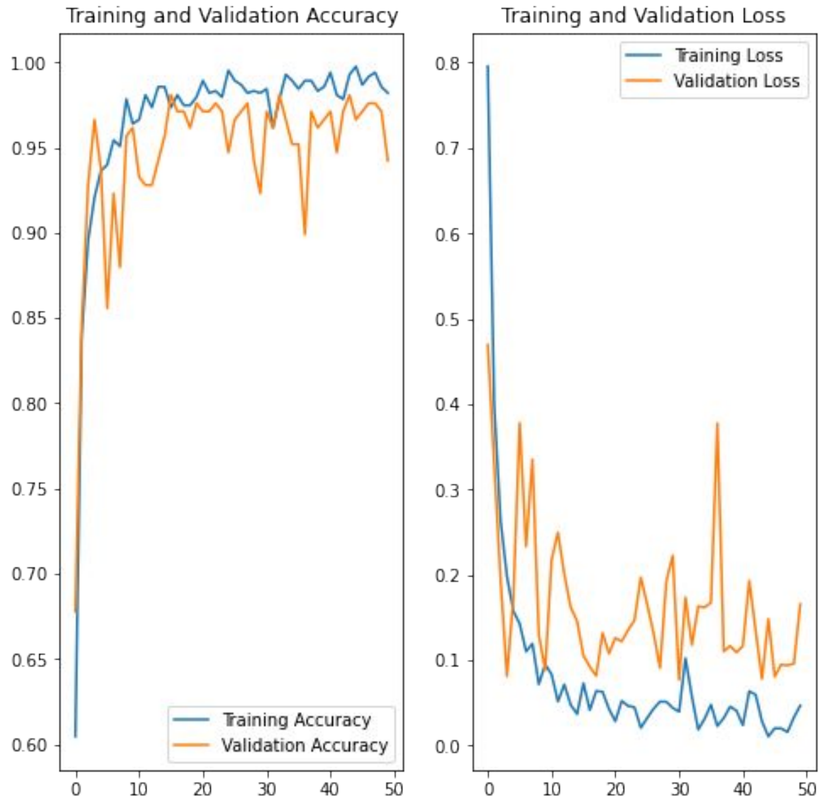
best model accuracy 0.95

Future Works:

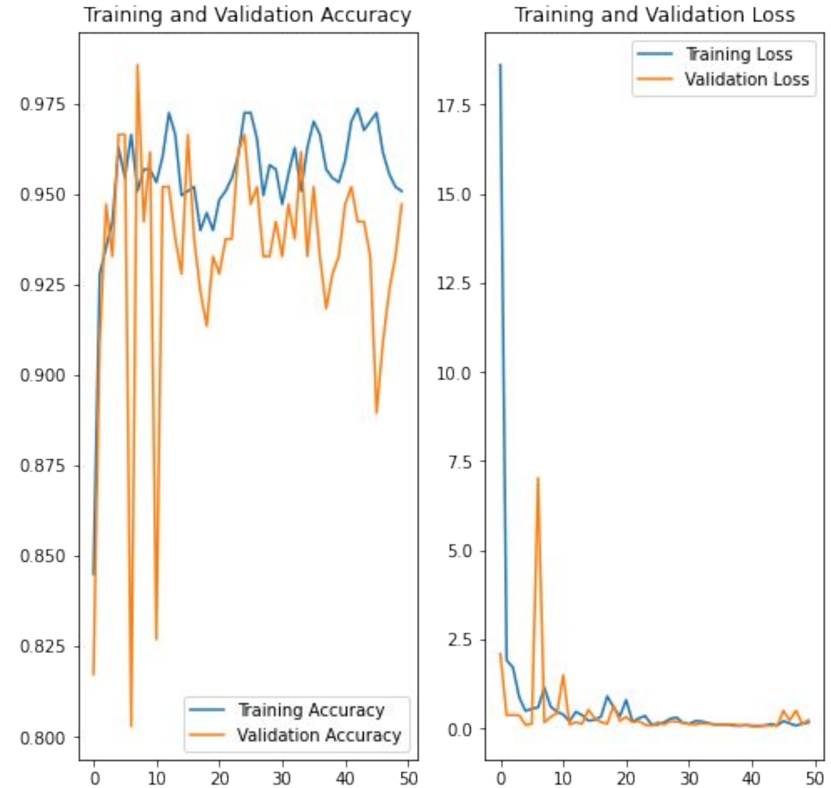
1) extend the dataset
2) multiclass classification (covid, pneumonia, pneumothorax ...)



Appendix: additional exp. fail plots (1)

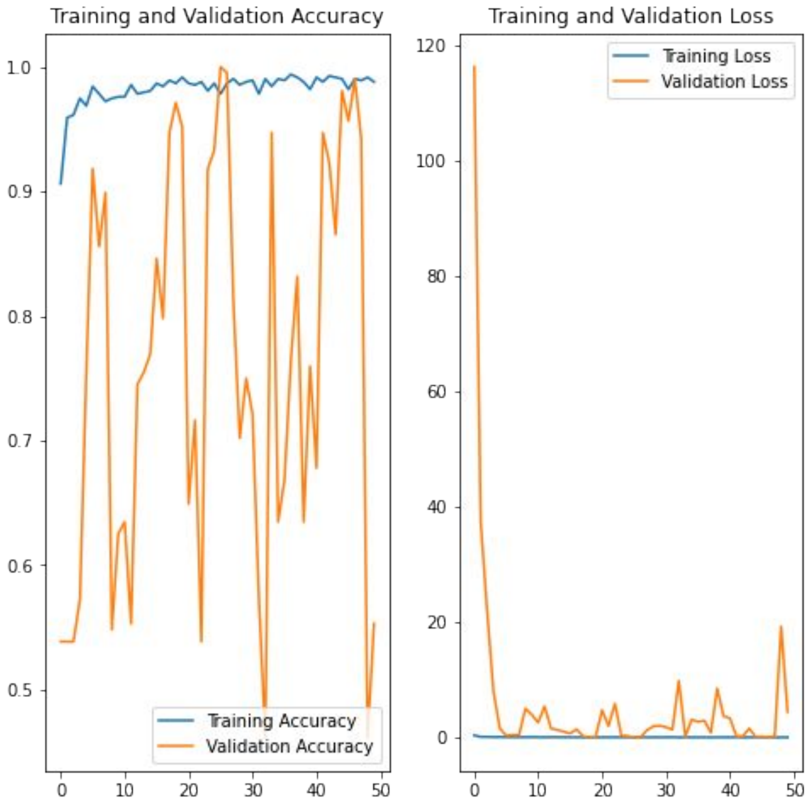


Conv2D: 16 32 64 128 256 512
No batch normalization, no LR decay
Very long training time
Good accuracy but unstable: bad inference

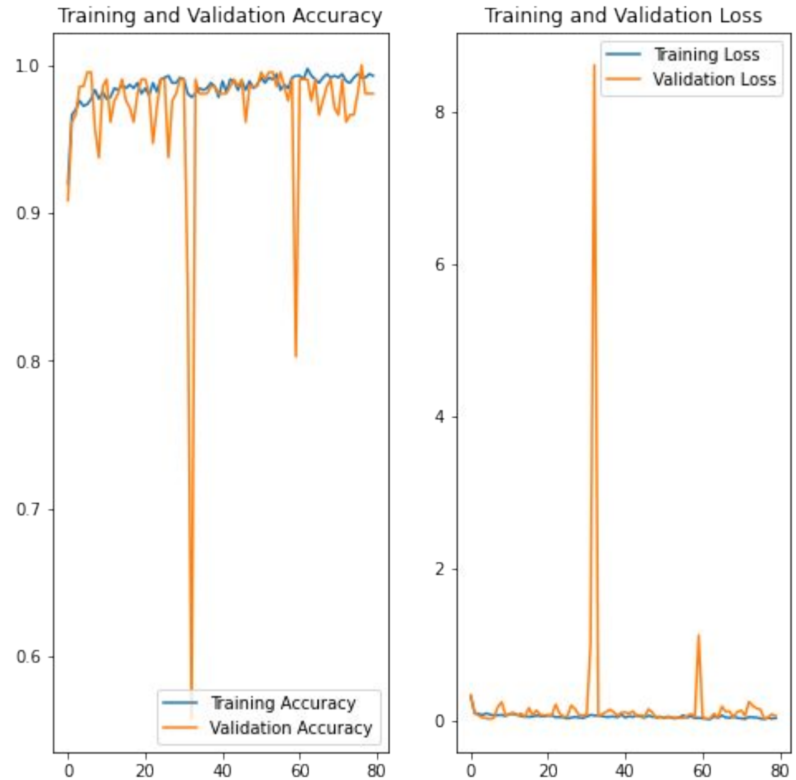


Conv2D: 16 32 64 128 256
First introduction of the batch norm
Very unstable training
Val_accuracy oscillating, bad inference

Appendix: additional exp. fail plots (2)



Deep network (32 32 64 64 … 512)
No data augmentation
Batch size small (8), big b.n. momentum (0.99)
LR: 0.1 without decay (very unstable net)



Standard network (shown before)
Batch norm before max pooling
No LR decay: instability
Inference not so bad

THANKS FOR YOUR ATTENTION