Kaggle Project

Detection of pneumonia in chest X-ray images using a deep learning approach

AUCOUTURIER Camille
BALOCHE Valentin

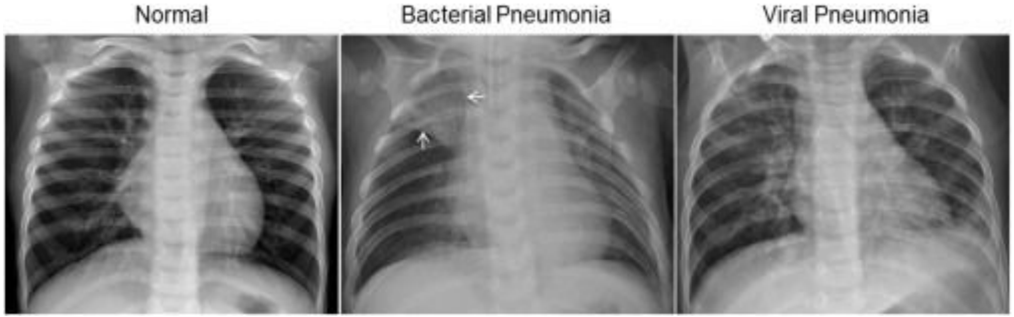
M2BI Learning Methods
2021/2022

Université
de Paris

Foreword

- Pneumonia = a form of acute respiratory infection that affects the lungs (*WHO*)
  Caused by bacteria, viruses or fungi

  → **Diagnosis** = Chest X-ray

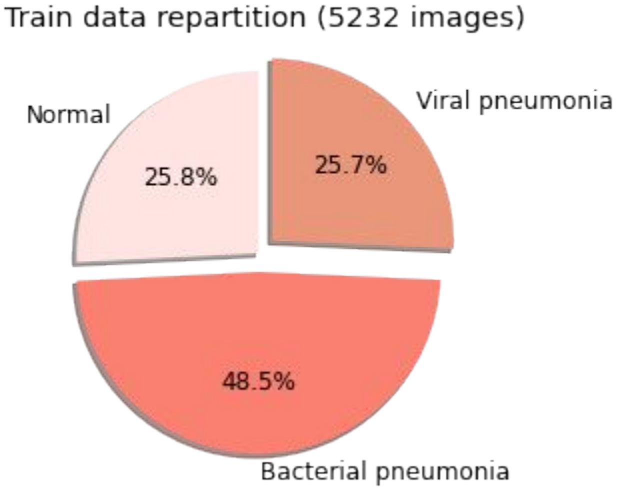


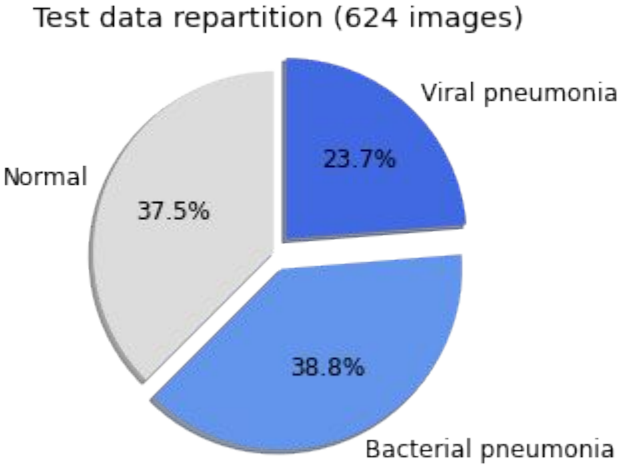
- **Objectif** : create a convolutional neural network (CNN) able to detect pneumonia from chest X-ray images

Data overview (1)

- Chest X-ray images from **retrospective cohorts of pediatric patients** of 1 to 5 years old from Guangzhou Women and Children’s Medical Center

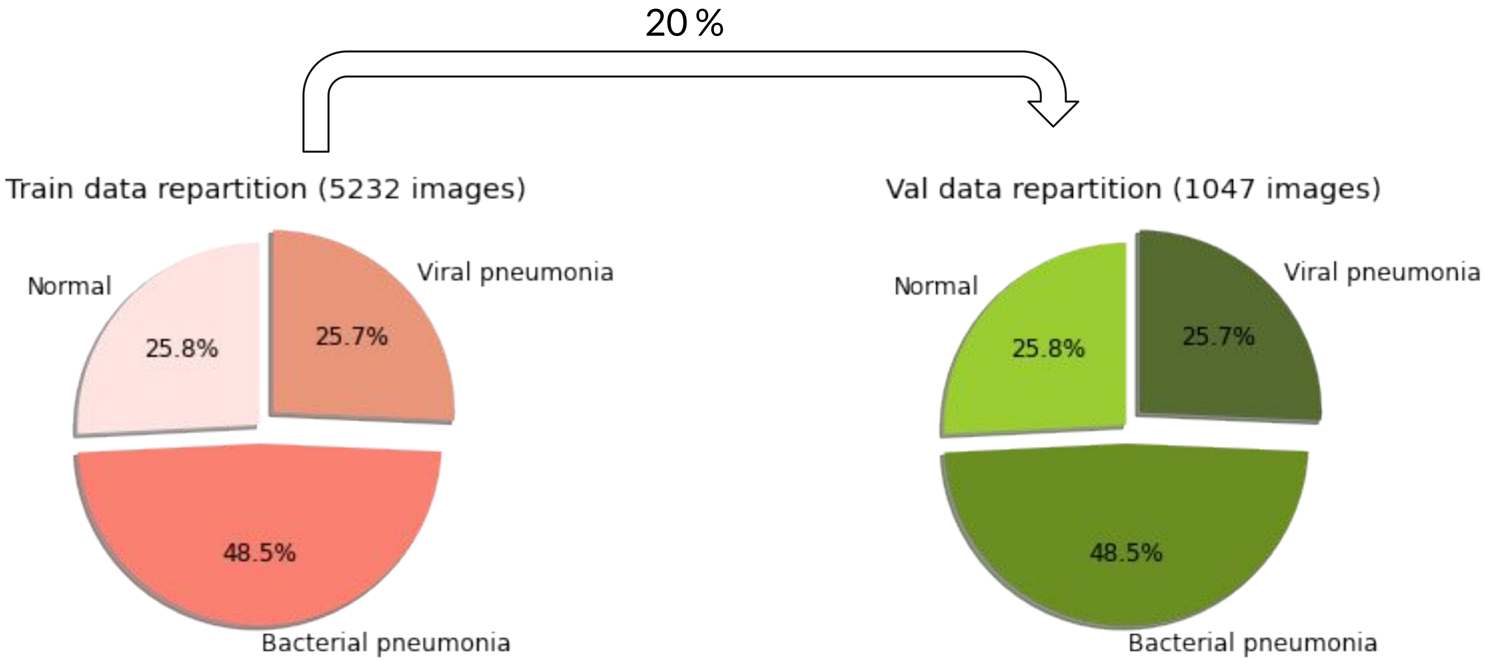
- **5856** images divided in 2 folders : train vs test





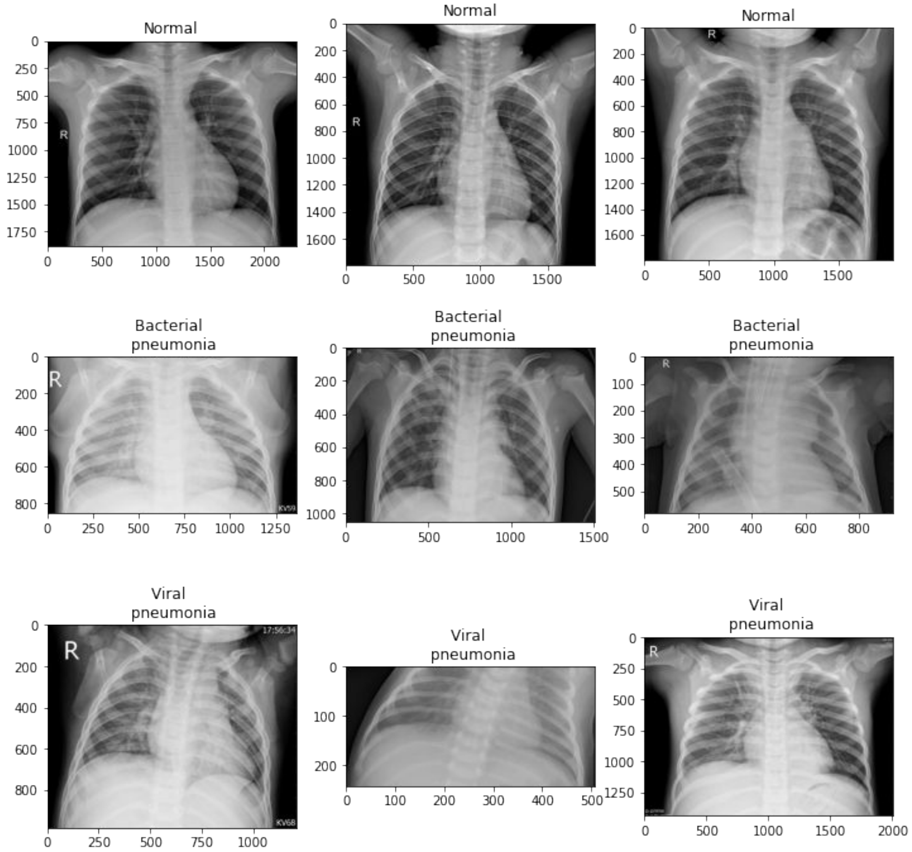
Data overview (2)

- Creation of a validation set

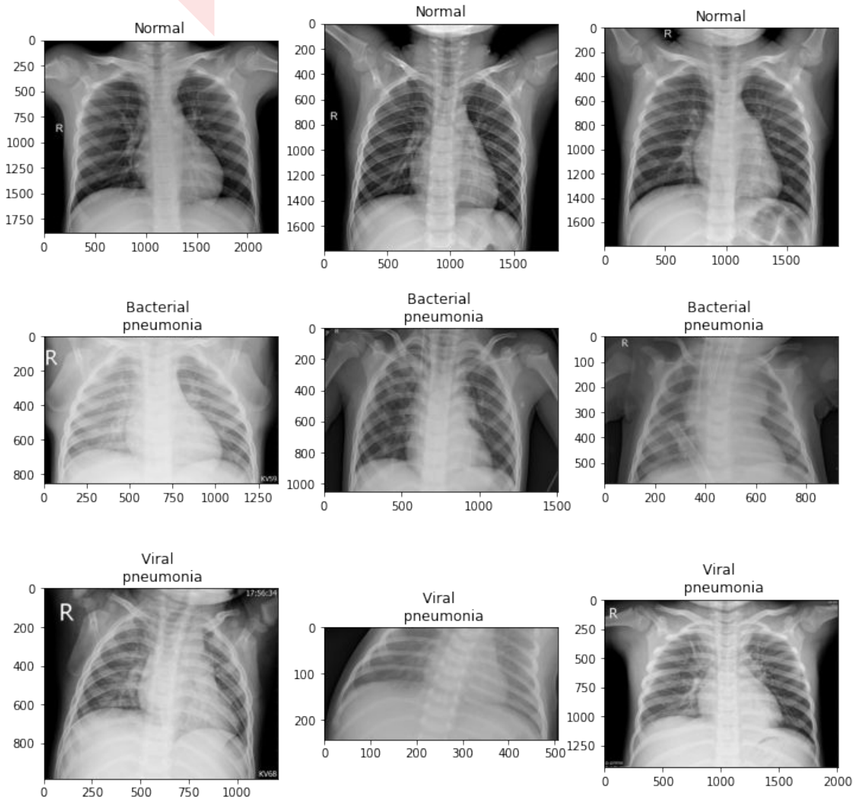


Data overview (3)

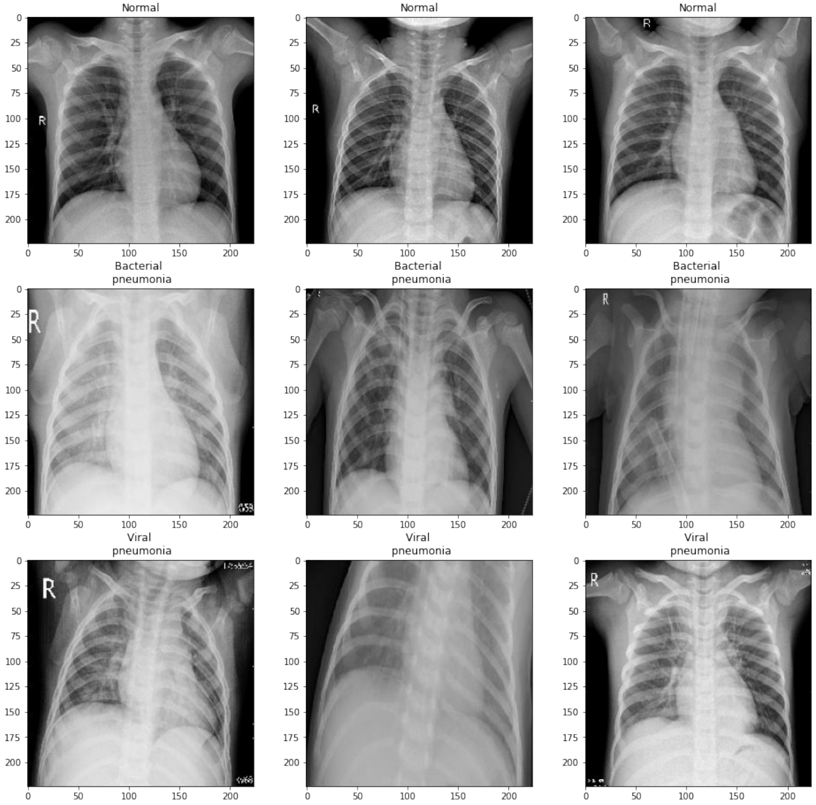
- Heterogeneous images shapes
- Needs to homogenize these data
- Target size ?
- BW or RGB ?



Resizing example



Resizing
224 x 224



Data preprocessing

Resizing images and selecting RGB format → Converting in array → Scaling to [0, 1] range

224 x 224 x 3

111	6	...	...	25	4
255	5	...	...	7	11
...	...	...	...	...	...
...	...	...	...	...	...
11	119	...	...	221	17
0	5	...	...	9	0

224

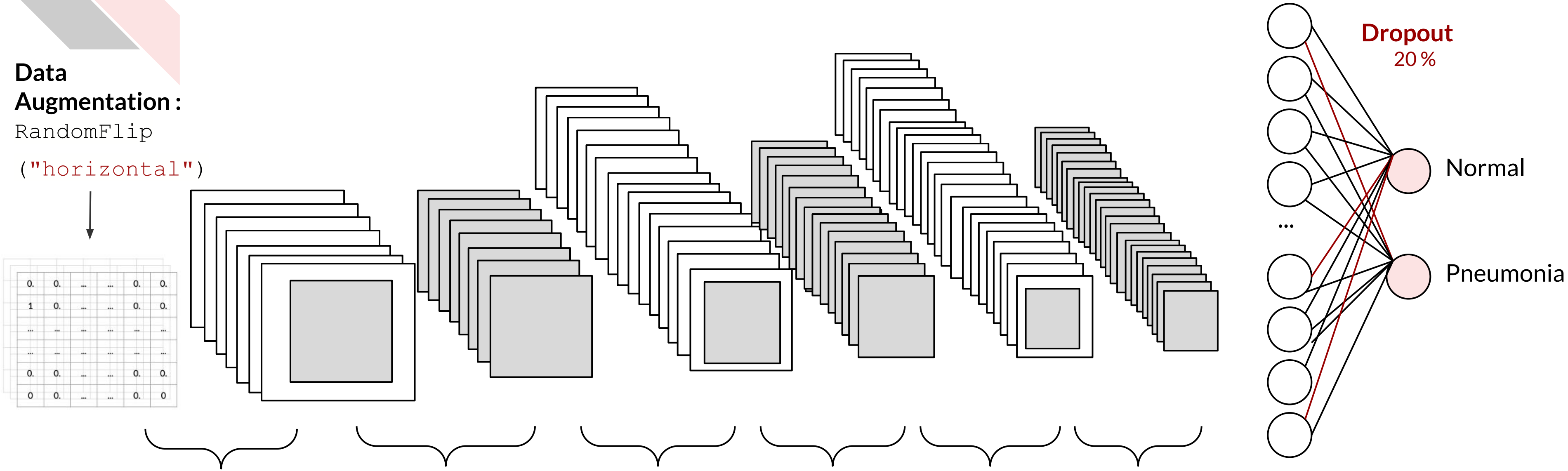
224

3

0.	0.	...	...	0.	0.
1	0.	...	...	0.	0.
...	...	...	...	...	...
...	...	...	...	...	...
0.	0.	...	...	0.	0.
0	0.	...	...	0.	0

CNN with 2 classes : Building

Data Augmentation :
`RandomFlip ("horizontal")`

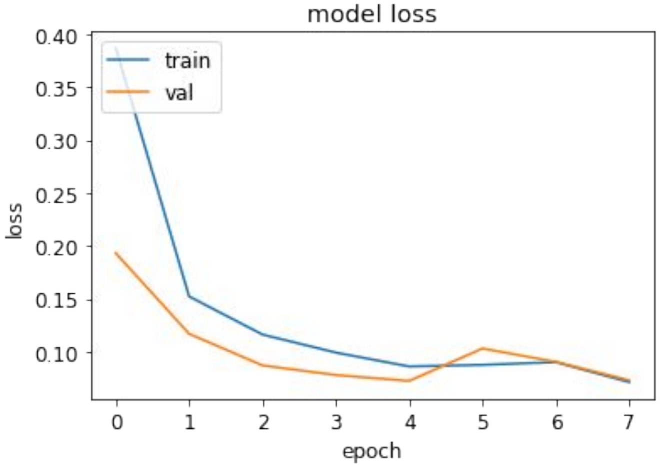


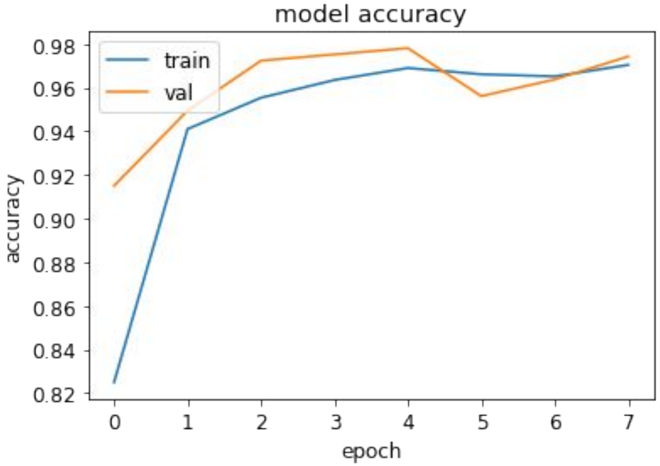
Input	Conv2D	MaxPooling	Conv2D	MaxPooling	Conv2D	MaxPooling	Flatten	Dense	**Output**
(224 x 224 x 3)	(222 x 222 x 3) 32 kernel (3,3) padding valid *relu*	(111 x 111 x 32) (2,2)	(109 x 109 x 64) 64 kernel (3,3) padding valid *relu*	(54 x 54 x 64) (2,2)	(52 x 52 x 128) 128 kernel (3,3) padding valid *relu*	(26 x 26 x 128) (2,2)	(86528)	(2) *softmax*	

```
Trainable params: 266,306
```

CNN with 2 classes : Training

- `batch_size = 32, epochs = 30`
- `callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3, verbose=0)`
- `opt = tf.keras.optimizers.Adam(learning_rate=0.0001, decay=1e-5)`
- `model_1.compile(loss="categorical_crossentropy", optimizer=opt, metrics=["accuracy"])`
- `history_1 = model_1.fit(x_2train, onehot_y_2train, batch_size, epochs, validation_data=(x_2val, onehot_y_2val),shuffle=True, callbacks=[callback])`





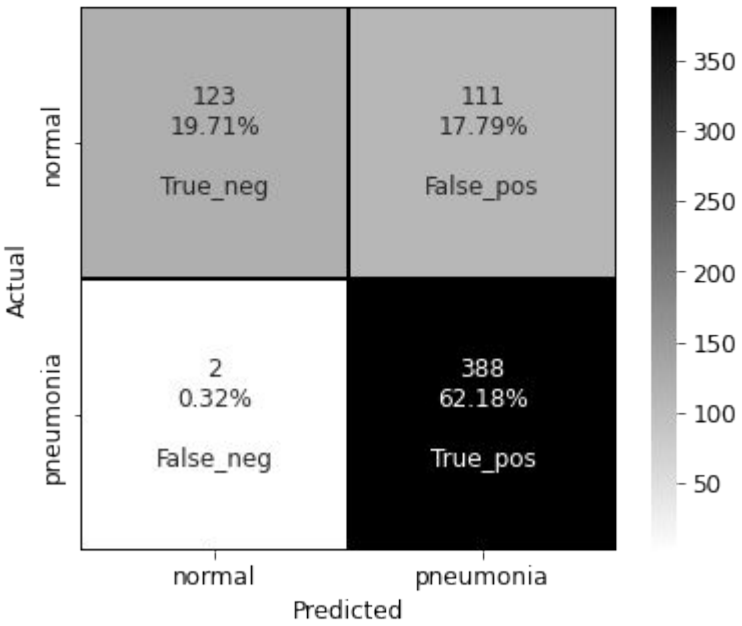
CNN with 2 classes : Evaluating

- Accuracy = 81, 891

- Good sensitivity but poor specificity

- Data distribution bias

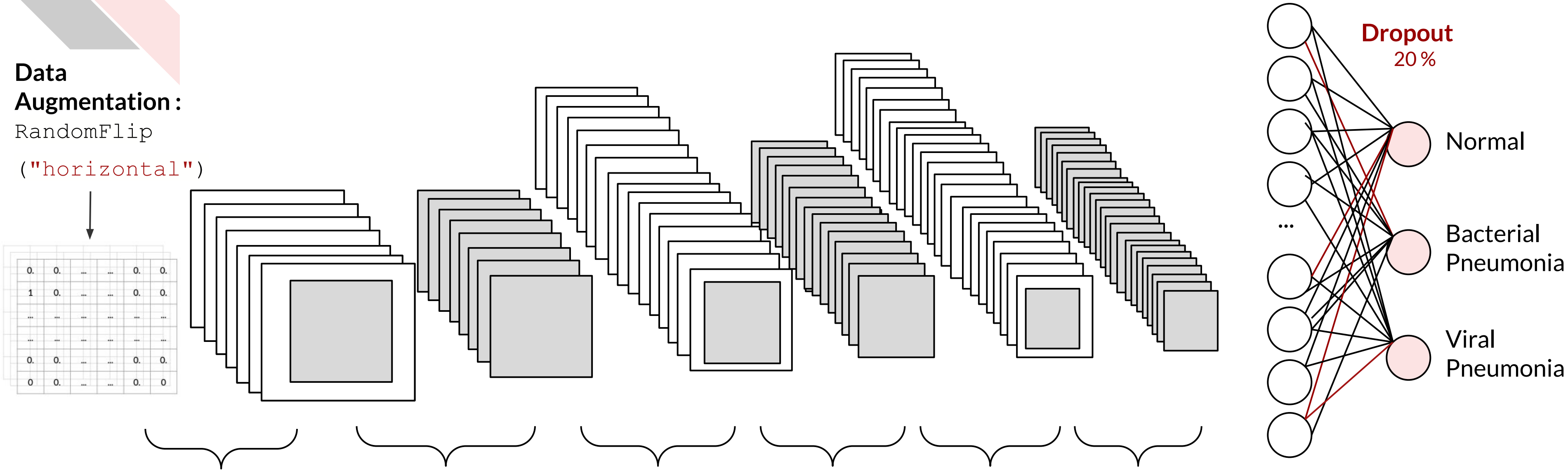
- Confusion matrix with test data :



Actual \ Predicted	normal	pneumonia
normal	123 19.71% True_neg	111 17.79% False_pos
pneumonia	2 0.32% False_neg	388 62.18% True_pos

CNN with 3 classes : Building

Data Augmentation :
`RandomFlip ("horizontal")`

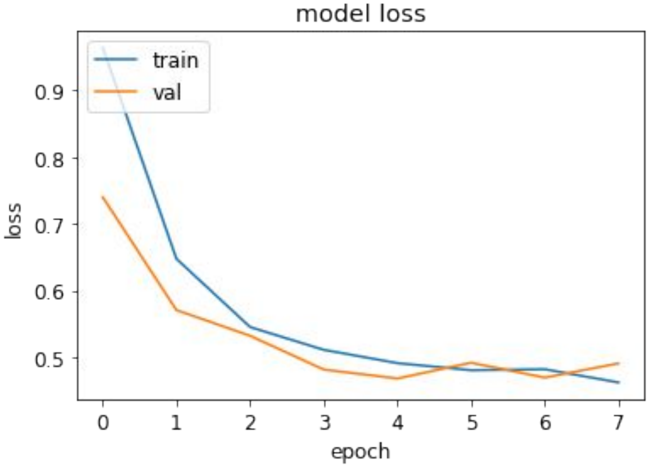


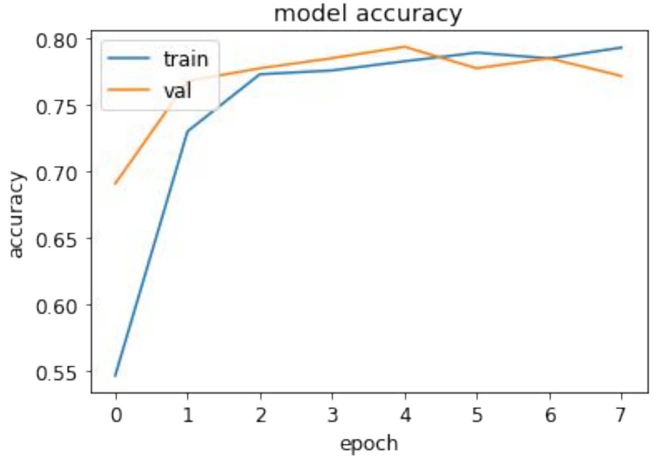
Input	Conv2D	MaxPooling	Conv2D	MaxPooling	Conv2D	MaxPooling	Flatten	Dense	**Output**
(224 x 224 x 3)	(222 x 222 x 3) 32 kernel (3,3) padding valid *relu*	(111 x 111 x 32) (2,2)	(109 x 109 x 64) 64 kernel (3,3) padding valid *relu*	(54 x 54 x 64) (2,2)	(52 x 52 x 128) 128 kernel (3,3) padding valid *relu*	(26 x 26 x 128) (2,2)	(86528)	(3) *softmax*	

```
Trainable params: 352,835
```

CNN with 3 classes : Training

- `batch_size = 64, epochs = 30`
- `callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3, verbose=0)`
- `opt = tf.keras.optimizers.Adam(learning_rate=0.0001, decay=1e-5)`
- `model_2.compile(loss="categorical_crossentropy", optimizer=opt, metrics=["accuracy"])`
- `history_2 = model_2.fit(x_3train, onehot_y_3train, batch_size, epochs, validation_data=(x_3val, onehot_y_3val),shuffle=True, callbacks=[callback])`





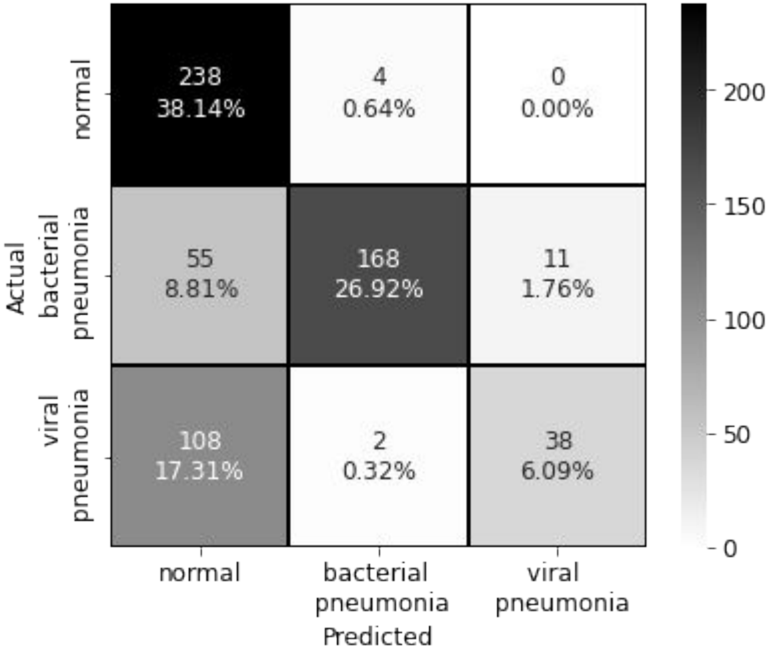
CNN with 3 classes : Evaluating

- **Accuracy = 71, 154**

- Good normal classification

- Difficulties in predicting viral pneumonia class

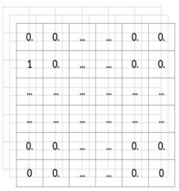
- Confusion matrix with test data :



Actual \ Predicted	normal	bacterial pneumonia	viral pneumonia
normal	238 38.14%	4 0.64%	0 0.00%
bacterial pneumonia	55 8.81%	168 26.92%	11 1.76%
viral pneumonia	108 17.31%	2 0.32%	38 6.09%

Transfer learning (1) : Building

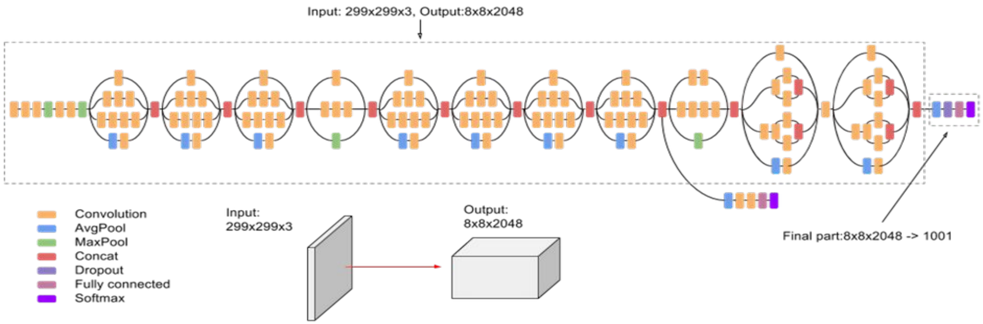
Data Augmentation : `RandomFlip("horizontal")`



Input

(224 x 224 x 3)

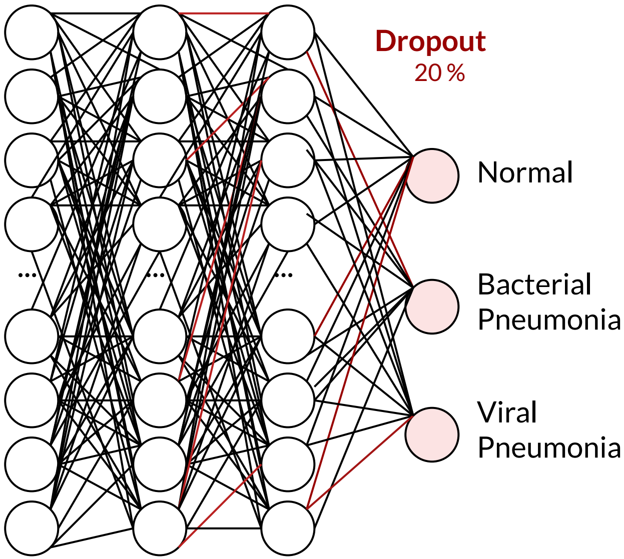
Pre trained model :
Inception V3



(5 x 5 x 2048)

```
base_model.layers.trainable=
False
```

Weights : "**imagenet**"



Dropout 20 %

Dropout 20 %

Normal

Bacterial Pneumonia

Viral Pneumonia

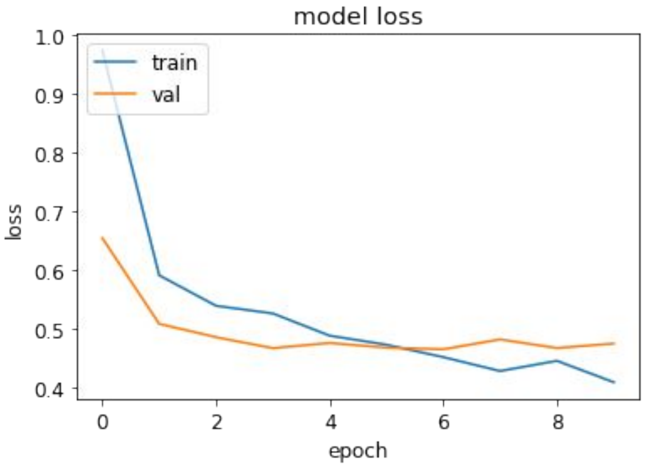
Flatten	Dense	Dense	Dense	**Output**
(51200)	(256)	(128)	(3)	
	relu	*relu*	*softmax*	

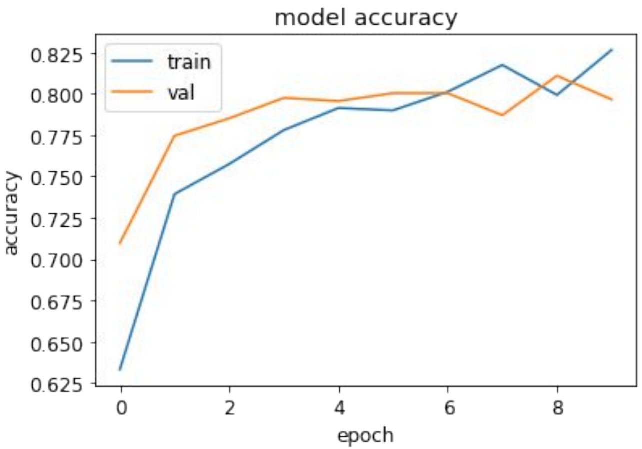
```
Non-trainable params : 21,802,784
```

```
Trainable params: 13,140,739
```

Transfer learning (1) : Training

```
batch_size = 64, epochs = 30
callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3, verbose=0)
opt = tf.keras.optimizers.Adam(learning_rate=0.0001, decay=1e-5)
model_3.compile(loss="categorical_crossentropy", optimizer=opt, metrics=["accuracy"])
history_3 = model_3.fit(x_3train, onehot_y_3train, batch_size, epochs,
validation_data=(x_3val, onehot_y_3val),shuffle=True, callbacks=[callback])
```





Transfer learning (1) : Evaluating

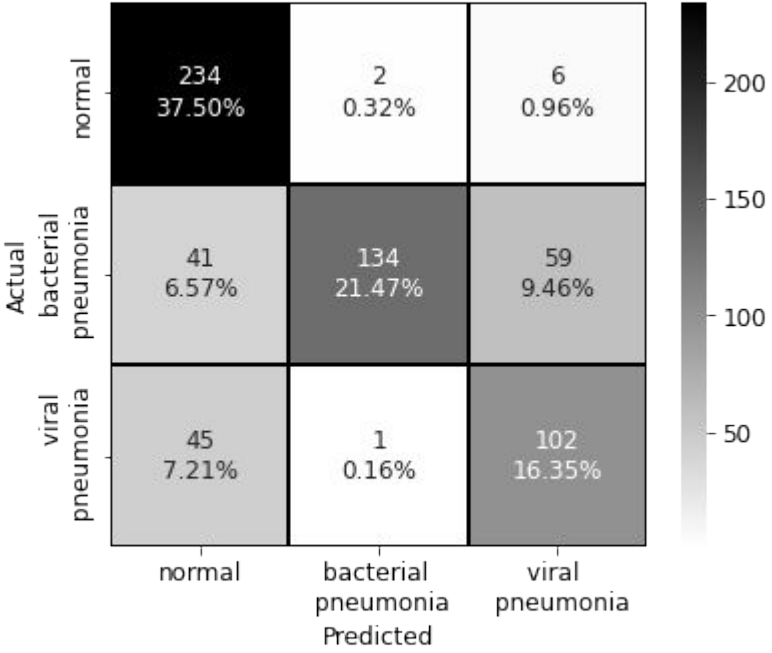
- Accuracy = 75, 321

- Viral pneumonia prediction improvement

- “Imagenet” not well adapted ?

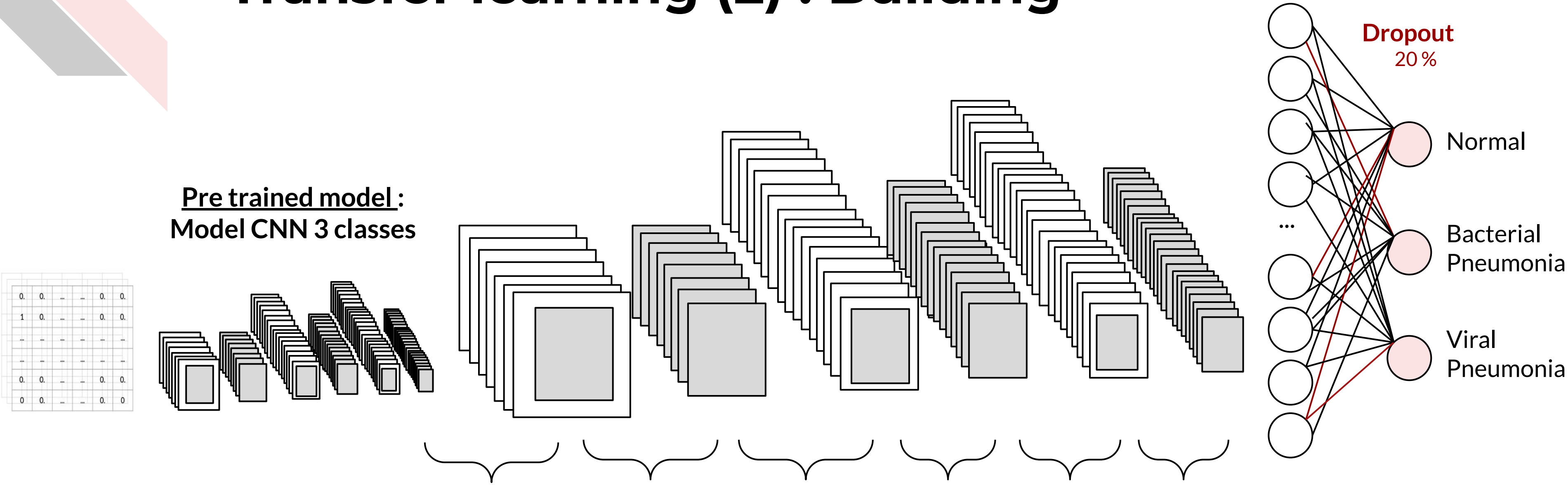
- Finetuning not much better...

- Confusion matrix with test data :



Actual \ Predicted	normal	bacterial pneumonia	viral pneumonia
normal	234 37.50%	2 0.32%	6 0.96%
bacterial pneumonia	41 6.57%	134 21.47%	59 9.46%
viral pneumonia	45 7.21%	1 0.16%	102 16.35%

Transfer learning (2) : Building



Pre trained model :
Model CNN 3 classes

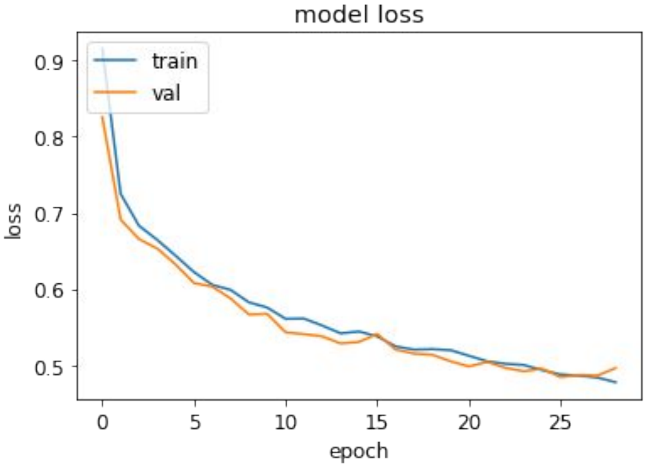
Input		Conv2D	Max Pooling	Conv2D	Max Pooling	Conv2D	Max Pooling	Flatten	Dense	**Output**
(224 x 224 x 3)	(26 x 26 x 128)	(24 x 24 x 32) 32 kernel (3,3) padding valid *relu*	(12 x 12 x 32) (2,2)	(10 x 10 x 64) 64 kernel (3,3) padding valid *relu*	(5 x 5 x 64) (2,2)	(3 x 3 x 128) 128 kernel (3,3) padding valid *relu*	(1 x 1 x 128) (2,2)	(128)	(3) *softmax*	

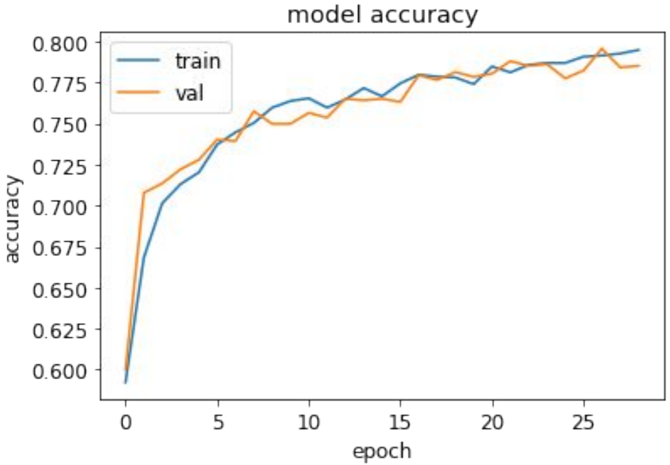
```
Non-trainable params : 93,248
```

```
Trainable params: 129,635
```

Transfer learning (2) : Training

- `batch_size = 32, epochs = 100`
- `callback = tf.keras.callbacks.EarlyStopping(monitor='val_loss', patience=3, verbose=0)`
- `opt = tf.keras.optimizers.Adam(learning_rate=0.0001, decay=1e-5)`
- `model_4.compile(loss="categorical_crossentropy", optimizer=opt, metrics=["accuracy"])`
- `history_4 = model_4.fit(x_3train, onehot_y_3train, batch_size, epochs, validation_data=(x_3val, onehot_y_3val),shuffle=True, callbacks=[callback])`

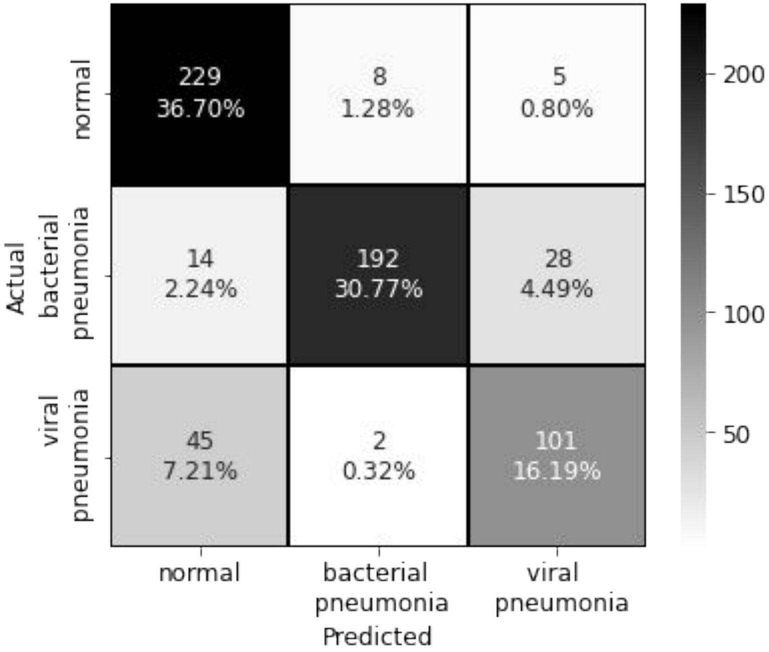




Transfer learning (2) : Evaluating

- Accuracy = 83, 654

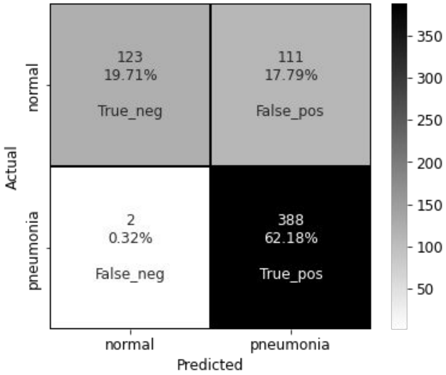
- Confusion matrix with test data :



Actual \ Predicted	normal	bacterial pneumonia	viral pneumonia
normal	229 36.70%	8 1.28%	5 0.80%
bacterial pneumonia	14 2.24%	192 30.77%	28 4.49%
viral pneumonia	45 7.21%	2 0.32%	101 16.19%

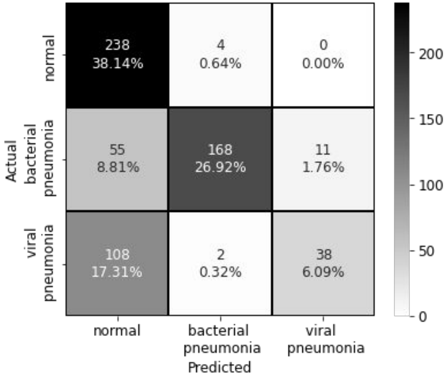
Conclusion

CNN - 2 classes



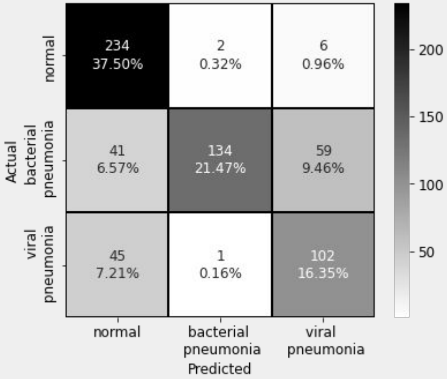
82%

CNN - 3 classes



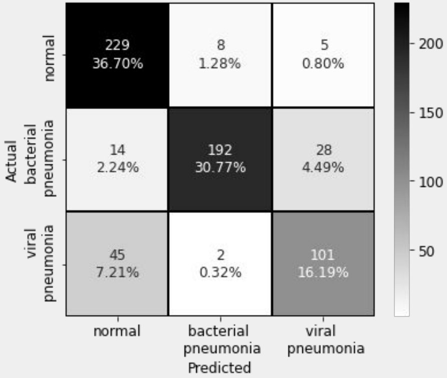
71%

Inception V3



75%

CNN_1



84%

TRANSFER LEARNING

Thank you for your attention !