

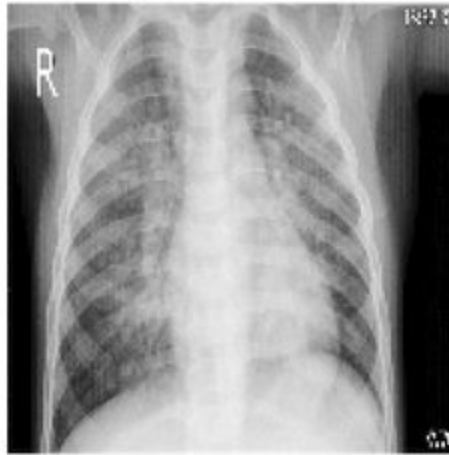


# Application de réseau de neurones à l'imagerie médicale:

## Radiographie pulmonaire

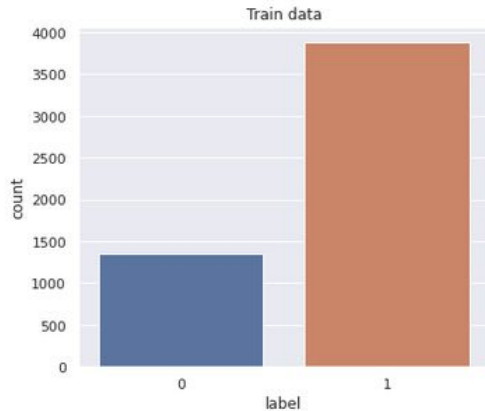
# Introduction et présentation des données

Objectif : Classification d'images de radiographie pulmonaire de patient (normal vs pneumonie)



# Introduction et présentation des données

- 2 datasets : Train ( 5228 images) et Test ( 624 images)
- Images redimensionnées (150x150) et coloré uniquement en gris
- Conversion des images en array et normalisation des valeurs de l'array
- Observation : déséquilibre entre les 2 classes à prévoir, 74% de “pneumonie”(1) dans le train



$$\text{weight\_0} = \frac{1}{n_{\text{normal}}} \times \frac{n_{\text{total}}}{2}$$

$$\text{weight\_1} = \frac{1}{n_{\text{pneumonie}}} \times \frac{n_{\text{total}}}{2}$$

# Méthode de recherche



- Implémentation de modèles de réseaux existants
- Réalisation de différents tests et modifications des modèles :
  - Paramètre de la Data augmentation
  - Structure des réseaux
  - Hyperparamètre (Learning Rate)
- Conservation des modèles les plus performants
- Transfert Learning (VGG, ResNet...)
- Comparaisons des modèles les plus performants

# Résultats

## 1. Meilleur modèle sans Data Augmentation (DA)

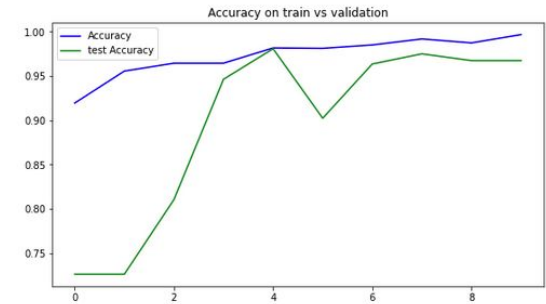
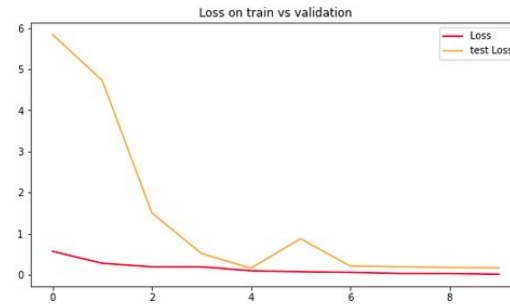
Layer (type)	Output Shape	Param #
input_14 (InputLayer)	[(None, 150, 150, 1)]	0
conv2d_52 (Conv2D)	(None, 148, 148, 16)	160
batch_normalization_52 (Batch Normalization)	(None, 148, 148, 16)	64
max_pooling2d_52 (MaxPooling2D)	(None, 74, 74, 16)	0
conv2d_53 (Conv2D)	(None, 72, 72, 32)	4640
batch_normalization_53 (Batch Normalization)	(None, 72, 72, 32)	128
max_pooling2d_53 (MaxPooling2D)	(None, 36, 36, 32)	0
conv2d_54 (Conv2D)	(None, 34, 34, 64)	18496
batch_normalization_54 (Batch Normalization)	(None, 34, 34, 64)	256
max_pooling2d_54 (MaxPooling2D)	(None, 17, 17, 64)	0
conv2d_55 (Conv2D)	(None, 15, 15, 128)	73856
batch_normalization_55 (Batch Normalization)	(None, 15, 15, 128)	512
max_pooling2d_55 (MaxPooling2D)	(None, 7, 7, 128)	0
flatten_13 (Flatten)	(None, 6272)	0
dense_26 (Dense)	(None, 32)	200736
dropout_13 (Dropout)	(None, 32)	0
dense_27 (Dense)	(None, 1)	33
Total params: 298,881		
Trainable params: 298,401		
Non-trainable params: 480		

### Modèle TF :

Obtient de bonnes performances sur le jeu de test après entraînement sans DA

Accuracy val : 0.95

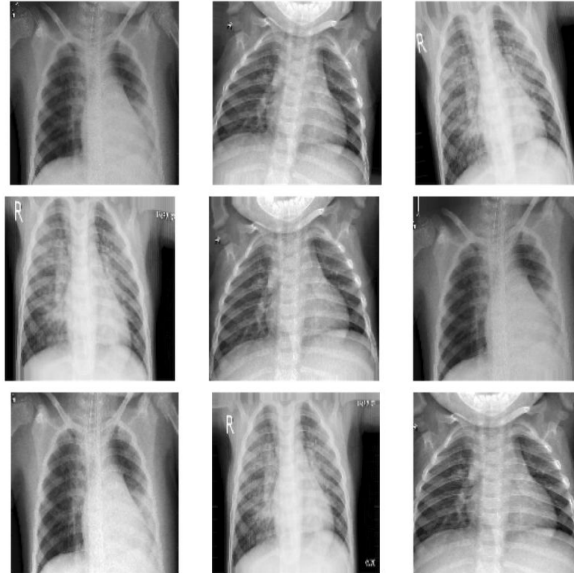
Accuracy test : 0.78



# Résultats

## 2. Recherche des meilleurs paramètres pour la DA

- Plusieurs paramètres sont testés un à un
  - Rotation\_range
  - Zoom\_range
  - Brightness\_range
  - Shear\_range
  - height\_range et width\_range



Rotation

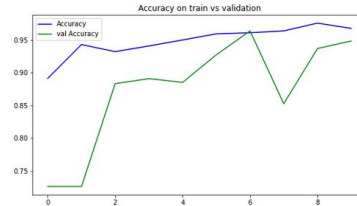
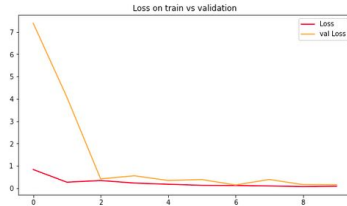
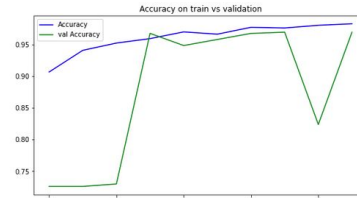
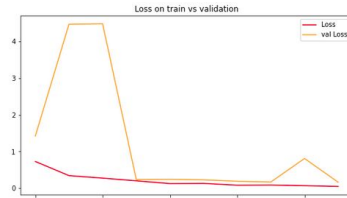
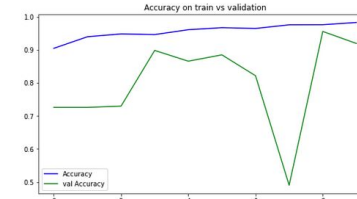
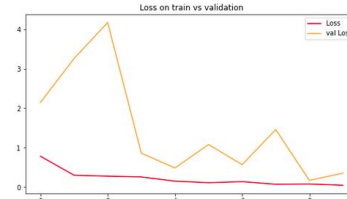
Déformation

Zoom

# Résultats

## 2. Recherche des meilleurs paramètres pour la DA

- Plusieurs paramètres sont testés un à un
- A la fin, 3 paramètres sont conservés :
  - la rotation des images
  - la déformation des images
  - le zoom sur les images



### Accuracy test

Rotation  
0.87

déformation  
0.87

Zoom  
0.85

# Résultats

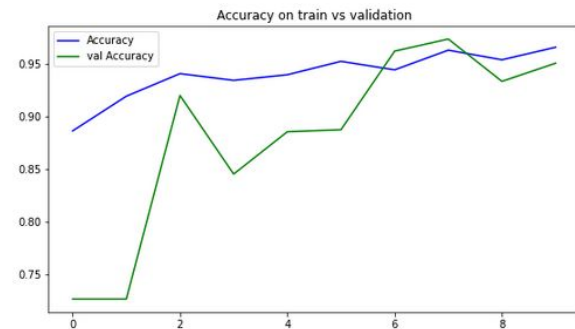
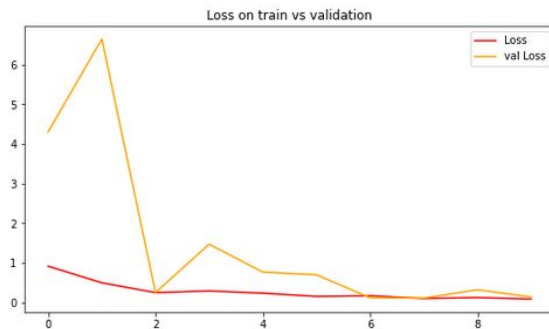
## 3. Data augmentation finale

Modèle simplifié auquel est ajouté la DA des données d'entraînement

Accuracy train : 0.956

Accuracy test :  $0.904 \pm 0.019$

92.4% de bonnes prédictions sur le test (après le meilleur entraînement)





# Résultats

## 4. Comparaison des différentes performances

	Accuracy test	Sensiti.(%)	Specifi.(%)	Number of Parameters
Model_TF	0.924	93.076923	91.452991	298 881
Model_DC	0.846	99.230769	60.256410	396 657
Model_K1	0.868	98.717949	67.094017	565 761
Model_DK	0.873	98.717949	68.376068	906 785
TL_VGG	0.877	87.435897	88.034188	14 722 881
TL_ResNet50V2	0.888	93.333333	81.196581	23 616 001
TL_InceptionV3	0.859	95.384615	70.085470	21 821 217
TL_DenseNet201	0.881	93.846154	78.632479	18 352 705

# Résultats.

## 5. Exemple de bonne et mauvaise prédiction

Predicted Class 1 | True Class 1    Predicted Class 0 | True Class 0



Predicted Class 0 | True Class 0    Predicted Class 1 | True Class 1



Predicted Class 0 | True Class 0    Predicted Class 1 | True Class 1



Predicted Class 0 | True Class 1    Predicted Class 1 | True Class 0



Predicted Class 0 | True Class 1    Predicted Class 1 | True Class 0



Predicted Class 0 | True Class 1    Predicted Class 1 | True Class 0



# Conclusion



- ★ Réseau de neurones capables de classer correctement la majorité des radiographies pulmonaires selon leur cas
- ★ Modèle avec relativement peu de couches et de paramètres à entraîner
- ★ Entraînement rapide du réseau (~3 minutes en GPU)
- ★ Performance globalement supérieure à celles de modèles avec plus de paramètres

## Poursuite possible :

- ❑ Possible améliorations supplémentaires du modèle
- ❑ Adapter le modèle pour différencier 3 classes (normal, pneumonie bactérienne et pneumonie virale)



**Merci pour votre attention**



Layer (type)	Output Shape	Param #
input_22 (InputLayer)	[(None, 150, 150, 1)]	0
conv2d_102 (Conv2D)	(None, 148, 148, 16)	160
conv2d_103 (Conv2D)	(None, 146, 146, 16)	2320
batch_normalization_79 (Batch Normalization)	(None, 146, 146, 16)	64
max_pooling2d_79 (MaxPooling)	(None, 73, 73, 16)	0
conv2d_104 (Conv2D)	(None, 71, 71, 32)	4640
conv2d_105 (Conv2D)	(None, 69, 69, 32)	9248
batch_normalization_80 (Batch Normalization)	(None, 69, 69, 32)	128
max_pooling2d_80 (MaxPooling)	(None, 34, 34, 32)	0
conv2d_106 (Conv2D)	(None, 32, 32, 64)	18496
conv2d_107 (Conv2D)	(None, 30, 30, 64)	36928
batch_normalization_81 (Batch Normalization)	(None, 30, 30, 64)	256
max_pooling2d_81 (MaxPooling)	(None, 15, 15, 64)	0
conv2d_108 (Conv2D)	(None, 13, 13, 128)	73856
conv2d_109 (Conv2D)	(None, 11, 11, 128)	147584
batch_normalization_82 (Batch Normalization)	(None, 11, 11, 128)	512
max_pooling2d_82 (MaxPooling)	(None, 5, 5, 128)	0

## Modèle DC

flatten_20 (Flatten)	(None, 3200)	0
dense_40 (Dense)	(None, 32)	102432
dropout_20 (Dropout)	(None, 32)	0
dense_41 (Dense)	(None, 1)	33
=====		
Total params: 396,657		
Trainable params: 396,177		
Non-trainable params: 480		

Layer (type)	Output Shape	Param #
input_28 (InputLayer)	[(None, 150, 150, 1)]	0
conv2d_136 (Conv2D)	(None, 144, 144, 16)	800
batch_normalization_96 (Batch Normalization)	(None, 144, 144, 16)	64
max_pooling2d_96 (MaxPooling)	(None, 72, 72, 16)	0
conv2d_137 (Conv2D)	(None, 66, 66, 32)	25120
batch_normalization_97 (Batch Normalization)	(None, 66, 66, 32)	128
max_pooling2d_97 (MaxPooling)	(None, 33, 33, 32)	0
conv2d_138 (Conv2D)	(None, 27, 27, 64)	100416
batch_normalization_98 (Batch Normalization)	(None, 27, 27, 64)	256
max_pooling2d_98 (MaxPooling)	(None, 13, 13, 64)	0
conv2d_139 (Conv2D)	(None, 7, 7, 128)	401536
batch_normalization_99 (Batch Normalization)	(None, 7, 7, 128)	512
max_pooling2d_99 (MaxPooling)	(None, 3, 3, 128)	0

## Modèle K1

flatten_23 (Flatten)	(None, 1152)	0
dense_46 (Dense)	(None, 32)	36896
dropout_23 (Dropout)	(None, 32)	0
dense_47 (Dense)	(None, 1)	33
Total params: 565,761		
Trainable params: 565,281		
Non-trainable params: 480		

Layer (type)	Output Shape	Param #
input_23 (InputLayer)	[(None, 150, 150, 1)]	0
conv2d_110 (Conv2D)	(None, 144, 144, 16)	800
conv2d_111 (Conv2D)	(None, 138, 138, 32)	25120
batch_normalization_83 (Batch Normalization)	(None, 138, 138, 32)	128
max_pooling2d_83 (MaxPooling2D)	(None, 69, 69, 32)	0
conv2d_112 (Conv2D)	(None, 65, 65, 32)	25632
conv2d_113 (Conv2D)	(None, 61, 61, 64)	51264
batch_normalization_84 (Batch Normalization)	(None, 61, 61, 64)	256
max_pooling2d_84 (MaxPooling2D)	(None, 30, 30, 64)	0
conv2d_114 (Conv2D)	(None, 28, 28, 64)	36928
conv2d_115 (Conv2D)	(None, 26, 26, 128)	73856
batch_normalization_85 (Batch Normalization)	(None, 26, 26, 128)	512
max_pooling2d_85 (MaxPooling2D)	(None, 13, 13, 128)	0

## Modèle DK

flatten_21 (Flatten)	(None, 21632)	0
dense_42 (Dense)	(None, 32)	692256
dropout_21 (Dropout)	(None, 32)	0
dense_43 (Dense)	(None, 1)	33
Total params: 906,785		
Trainable params: 906,337		
Non-trainable params: 448		