

KAGGLE

Présentation projet

Deep Learning

GRANGER Alexis
HUGON Alexandre

LAICH Mouna
FRAJ Rayan



Compétition Kaggle

Classification des maladies visibles sur les feuilles de manioc
Métrique : Accuracy

1

Les données

Description

Traitements



Description des données fournis

- 20 000 images
- Tailles diverses
- 5 labels numérotés de 0 à 4
- Images et labels dans un csv ou tfrecords

0 CBB	1087
1 CBSD	2189
2 CGM	2386
3 CMD	13157
4 Healthy	2577





Importer ces données

TFrecords

Avantages d'être plus rapide à importer et à traiter.

Déjà convertie en clés/valeurs.

Des difficultés à en créer des personnalisés

Images bruts

On fait nous même la liaison entre le csv et les images dans le dossier.

On a pleinement le contrôle sur les images.

Plus long à traiter



Augmenter les données

Enrichir son dataset :

- Rotation
- Déformation
- Zoom
- Symétrie

Utilisation de tensorflow avec ImageDataGenerator

Resize des images pour s'adapter aux contraintes matérielles.

-> 128x128 + couleurs car maladies parfois de même forme

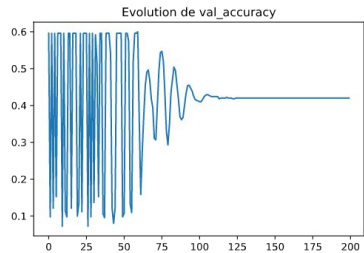
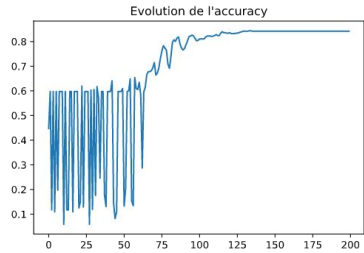
Modèles

Linéaire, CNN

Hyper-paramètres

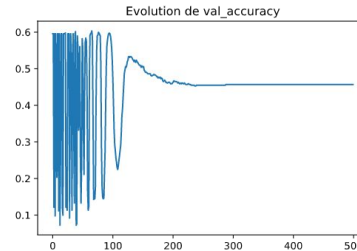
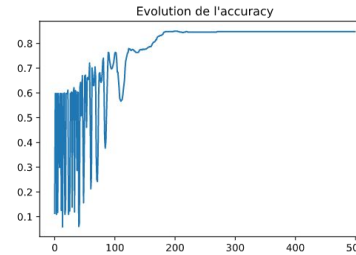


Linéaire



```
keras.optimizers.SGD(  
    lr=0.1, momentum=0.80),  
keras.losses.categorical_  
crossentropy,  
epochs=200,  
batch_size=512
```

Accuracy : 0.90 Val_accuracy : 0.41
Loss : 46.72 Val_loss : 558.48



```
keras.optimizers.SGD(lr=0  
.01, momentum=0.95),  
keras.losses.categorical_  
crossentropy,  
epochs=400,  
batch_size=512
```

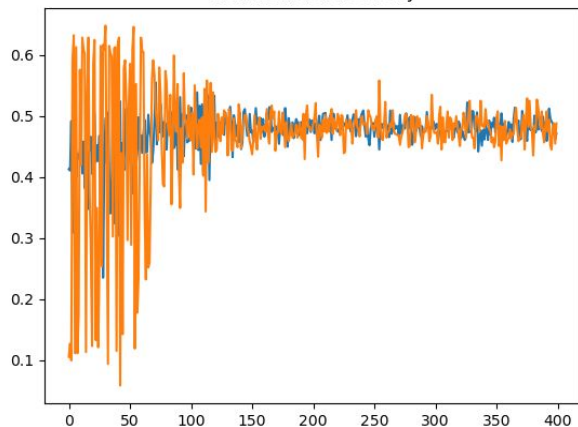
Accuracy : 0.84 Val_accuracy : 0.45
Loss : 88.32 Val_loss : 309.22



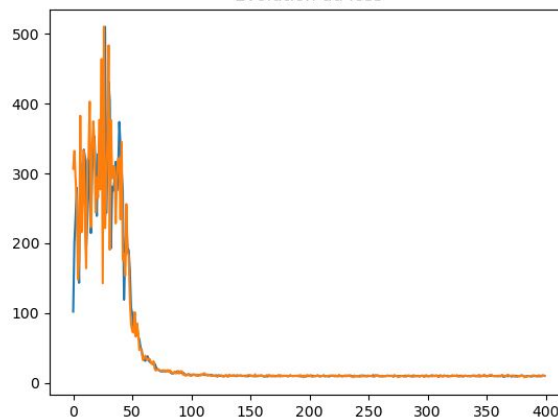
Linéaire

Avec données augmentées :

Evolution de l'accuracy



Evolution du loss

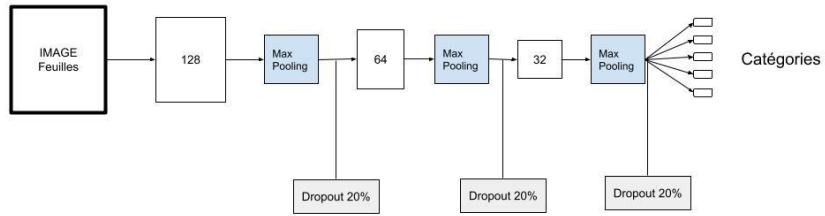


Accuracy : 0.47 Val_accuracy : 0.48
Loss : 1.25 Val_loss : 1.36

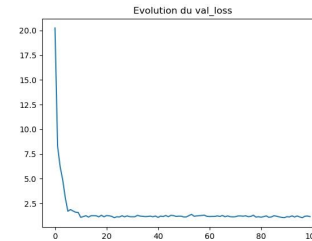
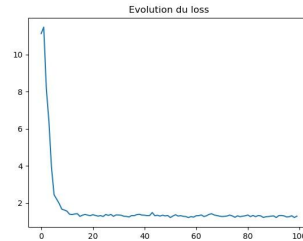
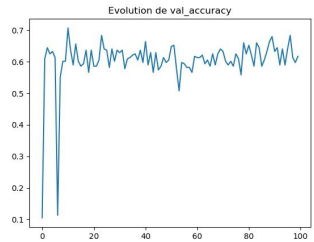
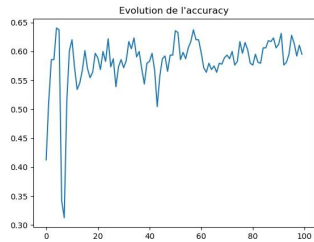
```
keras.optimizers.SGD(lr=0.01, momentum=0.95),  
keras.losses.categorical_crossentropy,  
epochs=400,  
batch_size=512,  
steps_per_epoch=20,  
val_steps=6
```



CNN Partie 1



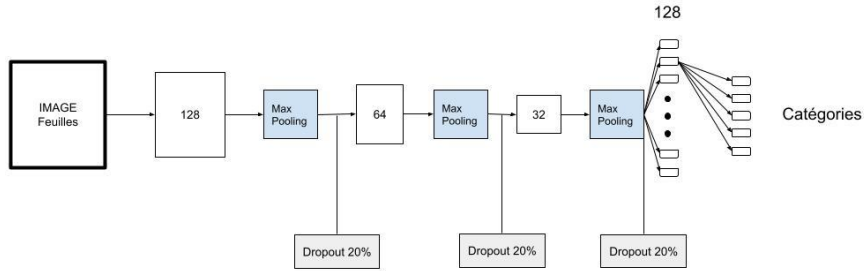
```
keras.optimizers.SGD(lr=0.01,  
momentum=0.95),  
keras.losses.categorical_crossentropy,  
epochs=100,  
batch_size=256,  
steps_per_epoch=10,  
val_steps=3
```



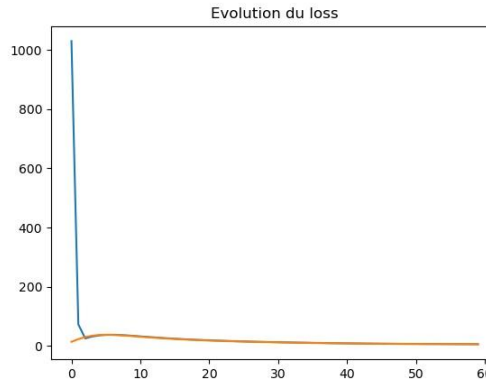
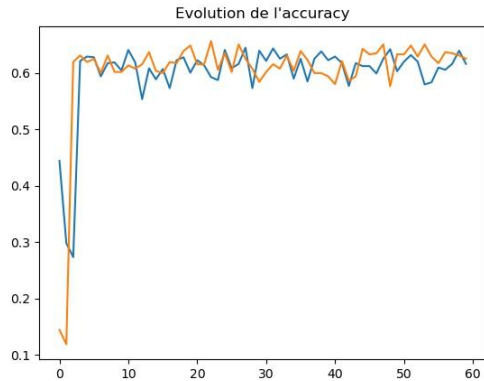
Accuracy : 0.61 Val_accuracy : 0.60
Loss : 1.62 Val_loss : 1.51



CNN Partie 2



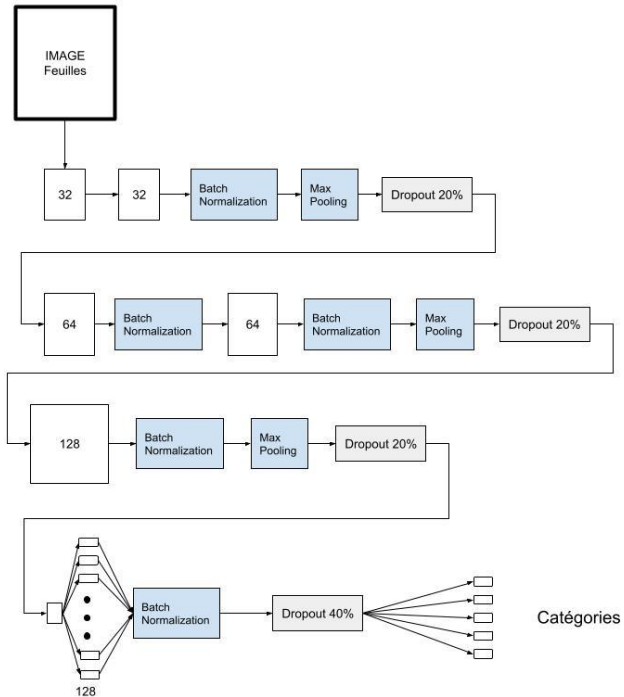
```
keras.optimizers.Adam(lr=0.01),  
keras.losses.categorical_crossentropy,  
epochs=60,  
batch_size=256,  
steps_per_epoch=20,  
val_steps=6
```



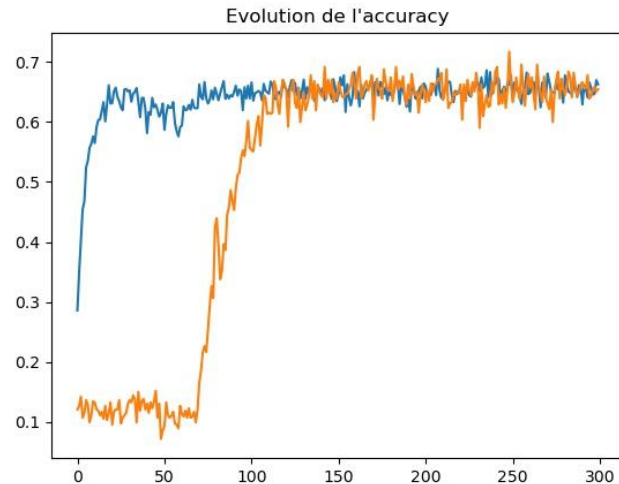
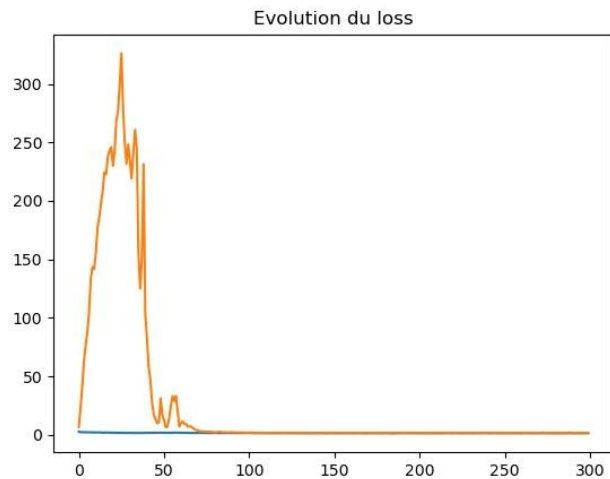
Accuracy : 0.62 Val_accuracy : 0.60
Loss : 1.65 Val_loss : 1.79



CNN Partie 3



```
keras.optimizers.Adam(lr=0.01),  
keras.losses.categorical_crossentropy,  
epochs=300,  
batch_size=256,  
steps_per_epoch=5,  
val_steps=2
```



Accuracy : 0.66 Val_accuracy : 0.65
Loss : 1.5 Val_loss : 1.58

Kaggle

Soumission

Score



Nos soumissions et scores

- Sauvegarde des modèles pré-entraînés
- Prédiction sur le dossier d'images "test"
- Export en CSV

Submission Cassava from preregistered Leaf (version 13/14) 2 days ago by Alexis From Notebook [Submission Cassava from preregistered Leaf]	Succeeded	0.602	<input type="checkbox"/>
---	-----------	-------	--------------------------

CNN Partie 2

Submission Cassava from preregistered Leaf (version 14/14) 2 days ago by Alexis From Notebook [Submission Cassava from preregistered Leaf]	Succeeded	0.608	<input type="checkbox"/>
---	-----------	-------	--------------------------

CNN Partie 3



CONCLUSION

CNN, EfficientNet pré-entraînés

Temps d'apprentissage