



Lieu : Total d'en bas Bafoussam

Site Web : www.keyce-it.fr

Email : a.akon@keyce-informatique.fr



**EXPOSE DE FRAMEWORK POUR
MACHINE LEARNING**

THEME :
**DETECTION D'OBJETS A L'AIDE DE
TENSORFLOW**

Rédigé et Présenté par :

❖ **DONFACK ERI CARNELLE**

Sous la direction de :

M. MINKOULOU Ulrich

Année 2023-2024

RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING

SOMMAIRE

SOMMAIRE	2
INTRODUCTION	3
I. CONTEXTE D'APPLICATION DE NOTRE TRAVAIL	4
II. ETAPES DE LA RÉALISATION DU PROJET	5
1. Préparation des Données	5
2. Analyse et Visualisation des Données :	5
3. Création du Modèle R-CNN	7
4. Entraînement du Modèle	8
5. Évaluation des Performances	9
6. Résultats et Analyse	10
CONCLUSION	12

RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING

INTRODUCTION

La classification d'images est l'une des tâches les plus courantes en vision par ordinateur. Ce rapport présente notre projet de classification d'images de chiens et de chats en utilisant des réseaux de neurones convolutionnels (CNN). L'objectif était de développer un modèle capable de distinguer automatiquement entre les images de chiens et de chats avec une précision élevée. Nous avons utilisé un jeu de données contenant des milliers d'images de chiens et de chats pour entraîner et tester notre modèle. Ce rapport détaille toutes les étapes du projet, y compris la préparation des données, la création du modèle, l'entraînement, l'évaluation des performances et les résultats obtenus.

I. CONTEXTE D'APPLICATION DE NOTRE TRAVAIL

La détection d'objets est un domaine de l'apprentissage automatique qui consiste à identifier et localiser les objets présents dans une image ou une vidéo. Cette tâche est essentielle pour de nombreuses applications telles que la vision par ordinateur, la conduite autonome, la surveillance, la robotique, etc.

Les principaux défis à relever pour développer un système de détection d'objets performant sont les suivants :

- ❖ **Représentation des données visuelles** : Concevoir des méthodes efficaces pour extraire et représenter les informations visuelles pertinentes (formes, textures, couleurs, etc.) à partir des images ou des vidéos.
- ❖ **Apprentissage des modèles de détection** : Développer des modèles d'apprentissage profond capables d'identifier et de localiser les objets de manière précise et robuste.
- ❖ **Gestion de la variabilité** : Concevoir des systèmes capables de fonctionner de manière fiable malgré la diversité des objets, des angles de vue, des conditions d'éclairage, etc.
- ❖ **Performances en temps réel** : Optimiser les modèles et les processus pour permettre une détection d'objets en temps réel, notamment pour les applications embarquées ou mobiles.

Cependant TensorFlow peut s'avérer très bénéfique pour le développement d'un système de détection d'objets raison pour laquelle nous allons faire usage de cette bibliothèque pour réaliser notre devoir car :

TensorFlow offre une vaste gamme de modèles de détection d'objets pré-entraînés (comme YOLO, Faster R-CNN, SSD, etc.) qui peuvent être facilement adaptés et affinés pour des cas d'utilisation spécifiques et nous allons utiliser Les R-CNN pour ce devoir.

II. ETAPES DE LA RÉALISATION DU PROJET

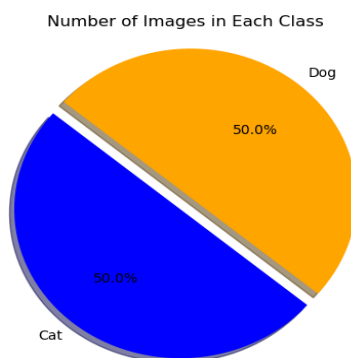
1. Préparation des Données

La première étape de notre projet consistait à préparer les données pour l'entraînement du modèle. Et pour obtenir les données on peut le faire de plusieurs façons :

- On peut soit nous-même créer nos propres données en effectuant une collecte des données.
- Utiliser les données pré-entraînées déjà existantes comme celle du géant Google avec sa bibliothèque de données COCO-Net (Common Objects in Context Network).
- Kaggle le site de compétition de la data science qui propose un éventail de données et c'est le choix que nous avons fait en téléchargeant le jeu de données des chiens et des chats puisque notre travail ici doit consister à détecter sur une image un objet lié au chien et au chat à la fois

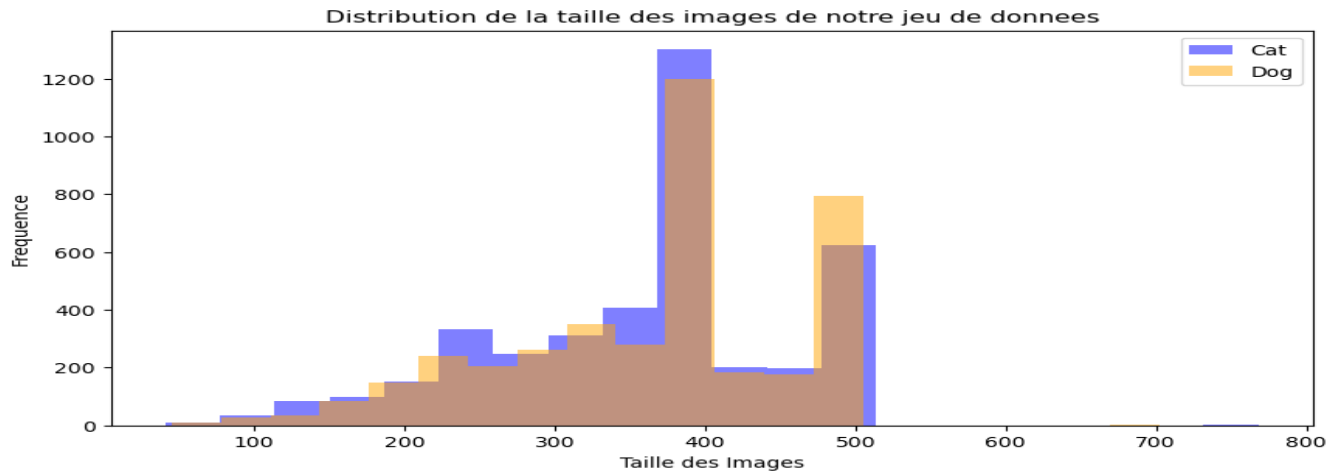
2. Analyse et Visualisation des Données :

La visualisation des données est une étape cruciale dans le développement de modèle de prédiction et même dans la science des données en général elle consiste à visualiser le comportement de notre jeu de données afin de les analyser et mieux les manier

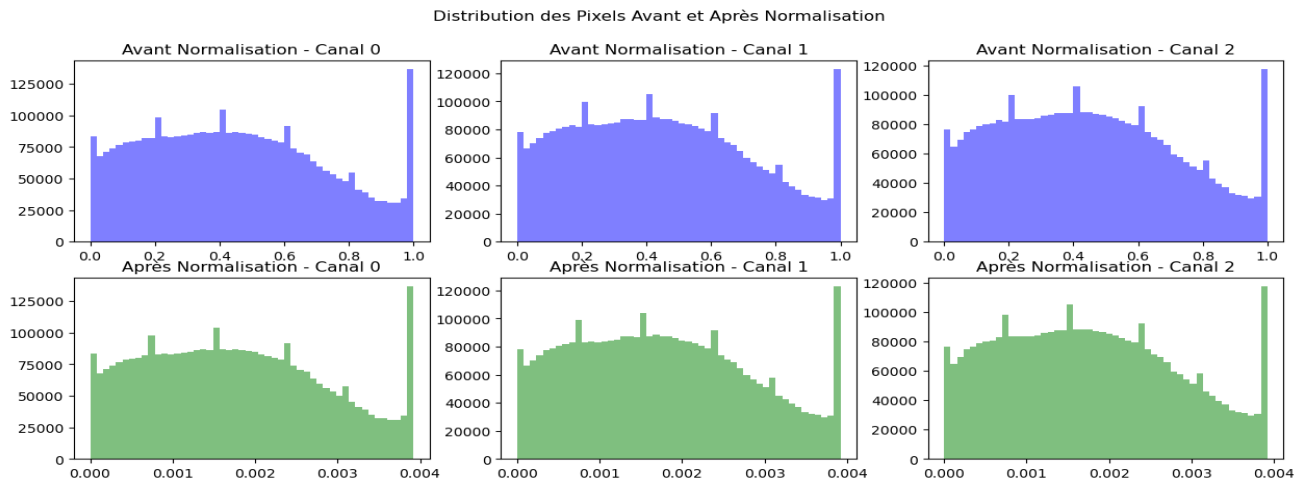


Le camembert ci présent nous montre la répartition de notre jeu de données et là on constate que notre jeu de données est équitablement réparti entre les deux variables que nous cherchons à détecter (cat et Dog) pour le cadre de ce projet

RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING



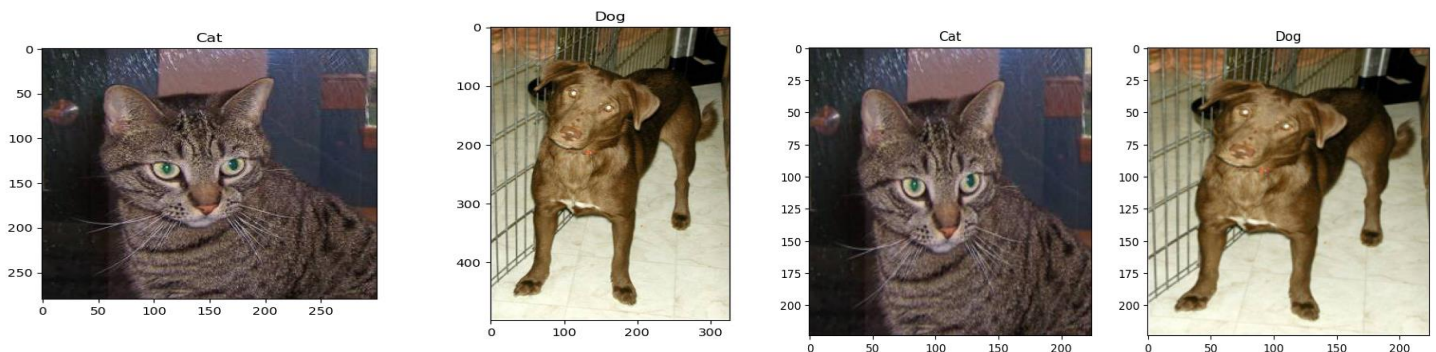
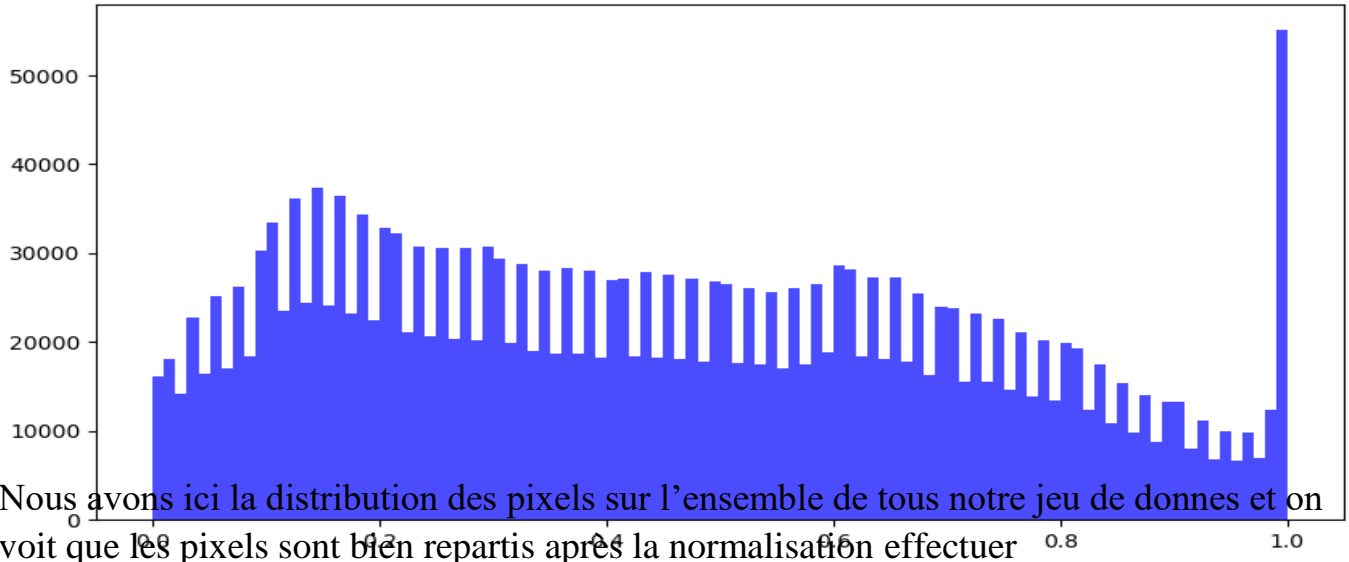
Cette visualisation nous aide à mieux comprendre la distribution des tailles d'images dans notre jeu de données, ce qui est très utile pour la suite du traitement et de l'entraînement de nos modèles de deep learning. Au regard de cette visualisation on peut dire que la distribution entre la taille des images liée au chat est presque répartie de la même façon que celle des chiens en termes de fréquences



Le schéma ci-dessus nous montre la distribution des pixels de nos différentes images du jeu de données. Il est nécessaire de toujours le vérifier afin de voir s'il y a une normalisation à faire. La normalisation consiste à placer toutes les valeurs de données à la même échelle (centrer la moyenne autour de 0 avec un écart type de 1) pour notre cas il s'agit des pixels de nos images. Une normalisation facilite le travail de notre modèle de prédiction car le modèle généralise beaucoup mieux avec les données normalisées et facilite également le processus de la fonction qui minimise les pertes. En bleu on a les données avant la normalisation et en vert après la normalisation.

RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING

Distribution des pixels des images d'entraînement



Echantillons d'une image de chien et de chat qui existe dans notre jeu de données avant et après la normalisation on constate donc que les pixels après la normalisation restent la même partout et n'est plus variable comme au départ

3. Création du Modèle R-CNN

A savoir : il faut noter qu'avant de passer à l'étape de la création de la couche de neurone il y a une étape que nous n'avons pas mentionner plus haut mais qui vaudrait mieux de savoir. Il s'agit entre autres d'une étape de prétraitement de la façon dont nos données doivent entrer dans la couche de convolution c'est à dire définir la taille d'entrer et autres ; on également appliquer autres transformations telle que le redimensionnement les zooms et les rotations qui constitue la fonction

RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING

de ImageGenerator de keras ceci dans le but d'optimiser le fonctionnement de notre couche de convolution afin d'optimiser au maximum ses performances.

Pour La construction de notre model Nous avons fait le choix des R-CNN (Region-based Convolutional Neural Network) plutôt qu'un CNN OU RNN qui ne peuvent prédire qu'une classe ce qui n'est pas le but de notre devoir car nous souhaitons faire une prédiction multi netiquette deux dans notre cas (Cat et ou Dog) et Les RCNN sont plus adapter car il est capable de prédire plusieurs labels. La sortie des paramètres de notre model ressemble donc à celle ci

Model: "model_1"		
Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 150, 150, 3)]	0
conv2d_5 (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d_5 (MaxPoolin g2D)	(None, 74, 74, 32)	0
conv2d_6 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_6 (MaxPoolin g2D)	(None, 36, 36, 64)	0
flatten_2 (Flatten)	(None, 82944)	0
dense_4 (Dense)	(None, 64)	5308480
dense_5 (Dense)	(None, 2)	130
=====		
Total params: 5328002 (20.32 MB)		
Trainable params: 5328002 (20.32 MB)		
Non-trainable params: 0 (0.00 Byte)		

4. Entraînement du Modèle

Le modèle a été entraîné sur l'ensemble d'entraînement à l'aide d'une descente de gradient stochastique (SGD). Nous avons utilisé la perte de catégorie croisée comme fonction de perte et avons ajusté les hyper paramètres tels que le taux d'apprentissage et le nombre d'époques (10) pour optimiser les performances du modèle. Nous avons également utilisé la validation croisée pour surveiller les performances du modèle pendant l'entraînement et éviter le su rajustement ou le sur apprentissage dont la sortie est celles présente par le schéma ci-dessous ou à une epoch de 10 on a

RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING

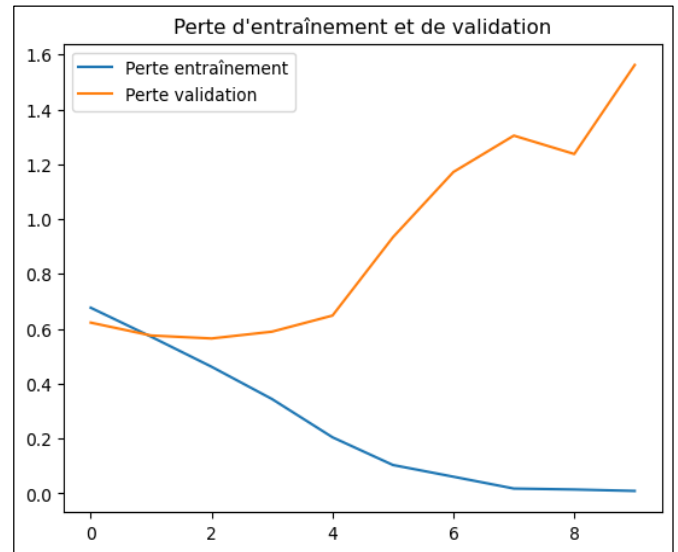
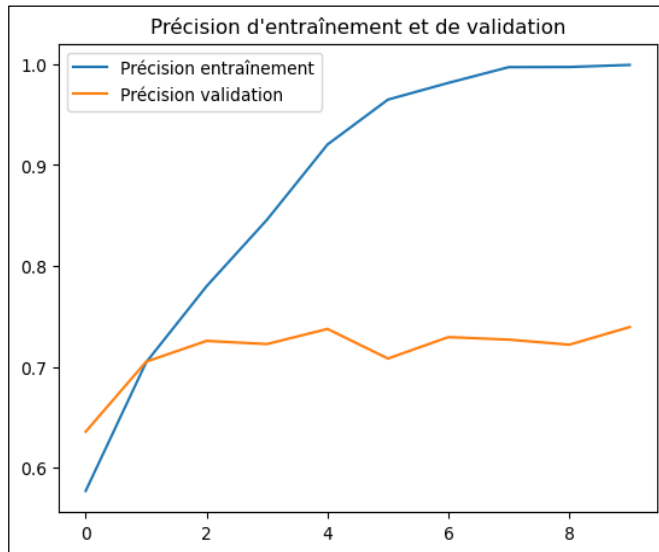
une précision de 99% et une perte de 0.01% et sur les données de validation avec une précision de 99% et la perte 1.2% on constate donc que par rapport à l'époch 9 la perte sur les données de validation a tendance à encore vouloir augmenter ce qui ne serait pas une bonne chose alors on aura donc le choix de sauvegarder notre modèle à un apprentissage de 9 epoch soit augmenter à 15 pour voir ce qui peut être amélioré

```
Epoch 1/10
200/200 [=====] - 95s 466ms/step - loss: 0.6768 - accuracy: 0.5769 - val_loss: 0.6225 - val_accuracy: 0.6356
Epoch 2/10
200/200 [=====] - 70s 347ms/step - loss: 0.5710 - accuracy: 0.7044 - val_loss: 0.5754 - val_accuracy: 0.7050
Epoch 3/10
200/200 [=====] - 70s 349ms/step - loss: 0.4615 - accuracy: 0.7798 - val_loss: 0.5646 - val_accuracy: 0.7256
Epoch 4/10
200/200 [=====] - 71s 356ms/step - loss: 0.3438 - accuracy: 0.8458 - val_loss: 0.5895 - val_accuracy: 0.7225
Epoch 5/10
200/200 [=====] - 72s 361ms/step - loss: 0.2044 - accuracy: 0.9203 - val_loss: 0.6481 - val_accuracy: 0.7375
Epoch 6/10
200/200 [=====] - 71s 352ms/step - loss: 0.1033 - accuracy: 0.9648 - val_loss: 0.9337 - val_accuracy: 0.7081
Epoch 7/10
200/200 [=====] - 72s 358ms/step - loss: 0.0606 - accuracy: 0.9814 - val_loss: 1.1715 - val_accuracy: 0.7294
Epoch 8/10
200/200 [=====] - 71s 356ms/step - loss: 0.0175 - accuracy: 0.9970 - val_loss: 1.3042 - val_accuracy: 0.7269
Epoch 9/10
200/200 [=====] - 73s 364ms/step - loss: 0.0140 - accuracy: 0.9972 - val_loss: 1.2372 - val_accuracy: 0.7219
Epoch 10/10
200/200 [=====] - 70s 348ms/step - loss: 0.0088 - accuracy: 0.9992 - val_loss: 1.5619 - val_accuracy: 0.7394
```

5. Évaluation des Performances

Une fois l'entraînement terminé, nous avons évalué les performances du modèle sur l'ensemble de test en calculant plusieurs métriques telles que la précision et la perte. et l'allure de notre courbe explique ce qu'on disait plus haut par rapport au résultat à l'époch 10

RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING



Pour pouvoir remédier à ce problème il sera question de jouer avec les hyper paramètres comme nous le disons soit s'arrêter à l'époch 9 soit augmenter à 15 et observer à chaque entraînement. Ce n'est pas seulement ces modifications qu'on peut effectuer on peut également regarder dans notre fonction qui nous permettait de construire le modèle et changer le type de perte utiliser par notre fonction de minimisation des erreurs réduire également le taux d'apprentissage pour permettre à notre stochastique gradient descend (SGD) de prendre plus de temps dans l'apprentissage. Mais pour notre cas le score de la précision nous satisfait déjà et nous allons donc utiliser ce modèle pour évaluer ses performances sur les prédictions.

6. Résultats et Analyse

Le modèle entraîné a obtenu une précision de 99%, une perte de 1%, sur l'ensemble de d'entraînement. Ce qui montre que le modèle est capable de bien généraliser aux données de test.

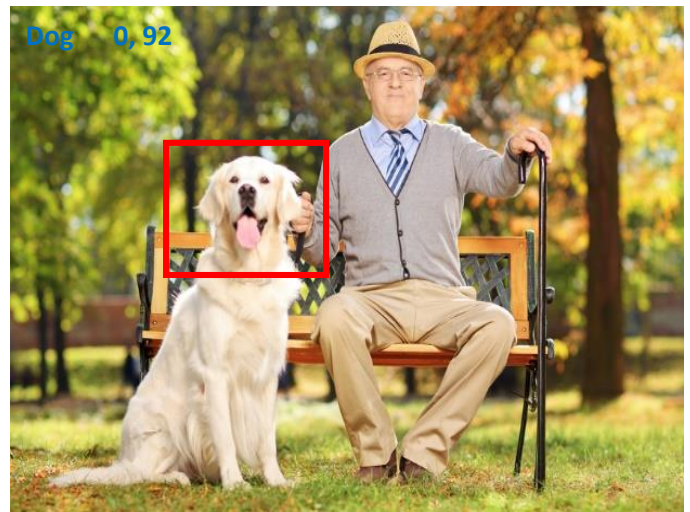
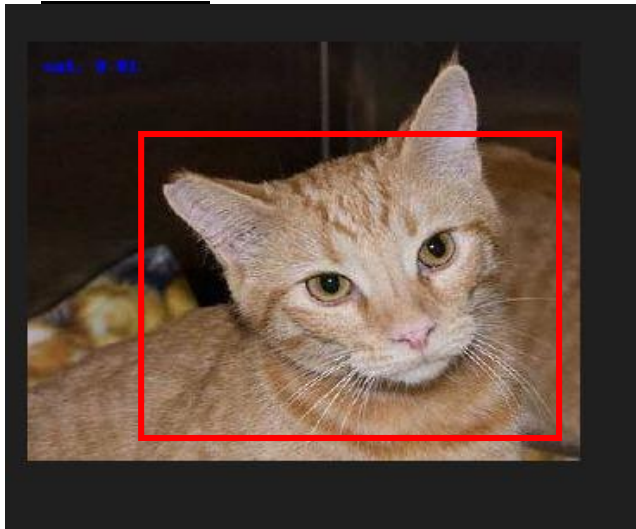
Pour confirmer cela nous a donc écrire une fonction qui doit charger le modèle entraîner sous l'extension. **Keras** prendre en entrée une image et retourner le résultat en ouvrant l'application qui nous permet de lire les images sur notre système et présenter le résultat sur l'image entrant

Nous avons donc pris deux images que nous allons tester et voir les prédictions de notre modèle avec la probabilité d'appartenance à la classe du label prédit

RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING



Résultats :



RAPPORT CC DE DEEP LEARNING ET FRAMEWORK MACHINE LEARNING

CONCLUSION

En conclusion, notre projet de classification d'images de chiens et de chats a été un succès. Nous avons développé un modèle R-CNN robuste qui est capable de distinguer avec précision entre les images de chiens et de chats. Ce modèle pourrait être utilisé dans une variété d'applications pratiques telles que la reconnaissance d'animaux de compagnie sur les réseaux sociaux, la surveillance vidéo, et bien d'autres encore. Ce projet démontre l'efficacité des réseaux de neurones convolutionnels pour la classification d'images et ouvre la voie à de futures recherches dans ce domaine.