Computer vision fundamentals

Wild cats detection and classification

Leo Škvorc

Zagreb, June 9. 2024.

Content

[1 Introduction 5](#_Toc169990745)

[2 Data 6](#_Toc169990746)

[3 Training models 7](#_Toc169990747)

[3.1 Custom model 7](#_Toc169990748)

[3.2 EfficentNetB0, ResNet50 and MobNet 7](#_Toc169990749)

[3.3 Training 7](#_Toc169990750)

[4 Results 8](#_Toc169990751)

[4.1 Accuracy loss f1 socres 8](#_Toc169990752)

[4.1.1 Accuracy 9](#_Toc169990753)

[4.1.2 Loss 10](#_Toc169990754)

[4.1.3 F1 score 11](#_Toc169990755)

[4.2 Confusion matrix 12](#_Toc169990756)

[4.4 Classification reports 13](#_Toc169990757)

[4.6 Image example 14](#_Toc169990758)

[5 Conclusion 15](#_Toc169990759)

Tables

**No table of figures entries found.**

Images

[Figure 1 Images spread 6](#_Toc169990734)

[Figure 2 Model scores 8](#_Toc169990735)

[Figure 3 Accuracy graph 9](#_Toc169990736)

[Figure 4 Loss graph 10](#_Toc169990737)

[Figure 5 F1 score graph 11](#_Toc169990738)

[Figure 6 Confusion matrix 12](#_Toc169990739)

[Figure 7 Custom model classification report 13](#_Toc169990740)

[Figure 8 ResNet model classification report 13](#_Toc169990741)

[Figure 9 MobNet model classification report 13](#_Toc169990742)

[Figure 10 EffNet model classification report 13](#_Toc169990743)

[Figure 11 Test prediction visualization 14](#_Toc169990744)

# Introduction

The domain of computer vision has advanced significantly, enabling sophisticated models that can identify and classify objects within images and videos. This project targets the challenge of detecting and classifying cats in wild settings, a task complicated by the variability and complexity of outdoor environments. The goal is to develop a robust and accurate model that can effectively identify and classify cats in diverse conditions. The developed model will then be compared to other 3 state of the art models: EfficientNetB0, ResNet50 and MobileNet. The goal is to determine what would be the best approach for training the model for cat classification.

# Data

Training and validation data is sourced from [10 Big Cats of the Wild – Image Classification](https://www.kaggle.com/datasets/gpiosenka/cats-in-the-wild-image-classification/data). The data is brought from train, validation and test folder to one master folder and then split 80-20 to train and valid data. This resulted in training data containing between 190 and 198 images for each class and validation data containing between 48 and 50 images for each class. Additionally test data is created by [Canva](https://www.canva.com) using prompts “(Cat name) in the wilderness”. Each test class has 40 images of different styles.

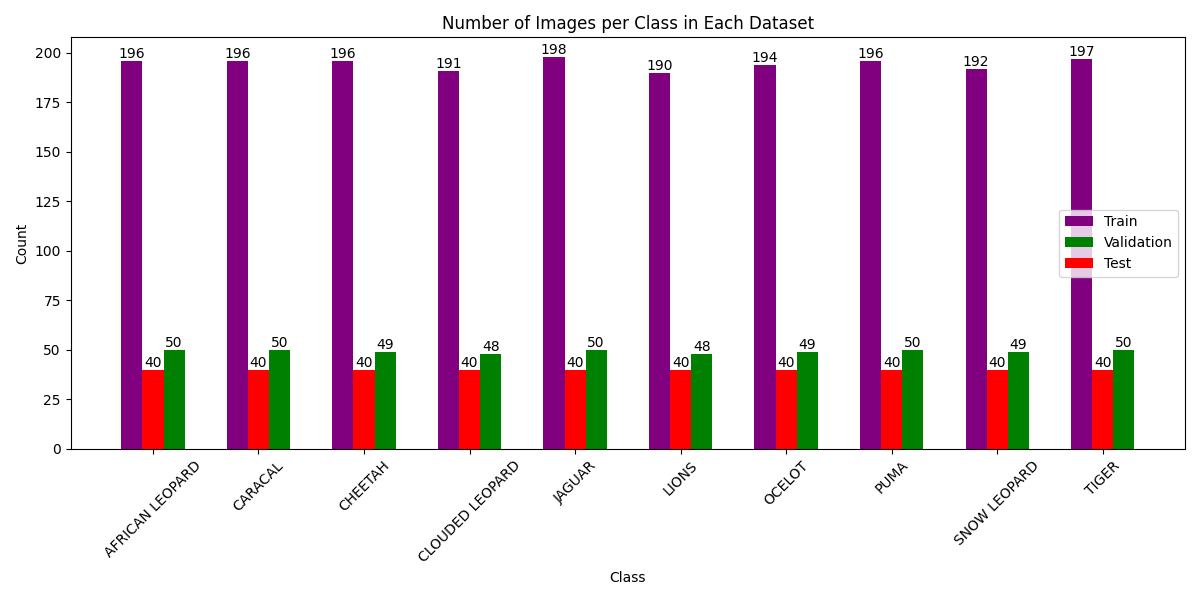


Figure 1 Images spread

To prepare the data for training image generators have been used to rescale all the images to 0 and 1. Additionally train images have randomly been rotated by 20 degrees, shifted by 20% and horizontally flipped.

Image generators have been created with respected datasets, targeted image size of 256x256, batch size of 16 and categorical class model.

Training data contains 1946 images, validation data contains 493 images and test data contains 400 images. The data is split into 10 classes: african leopard, caracal, cheetah, clouded leopard, jaguar, lions, ocelot, puma, snow leopard and tiger.

# Training models

In this project, four models were tested for detecting and classifying cats in wild settings. The first model is custom-made, while the remaining three are state-of-the-art models: ResNet50, MobileNet, and EfficientNetB0.

## Custom model

Custom model has been created using multiple convolutional and pooling layers. The first layer has an input shape of 256x246x3, representing the height, width and RGB colour channels of the images. All convolution layers utilize ReLU activation functions. Before the end final layer, the data is flattened and then passed through dense layers. The final dense layer outputs the number of classes with softmax activation function, allowing the model to predict the class of the given image.

## EfficentNetB0, ResNet50 and MobNet

The state-of-the-art models EfficientNetB0, ResNet50, and MobileNet were adjusted to fit the desired classes by adding pooling and dense layers. These modifications ensure that the models can handle the specific classification tasks required for this project while leveraging the powerful feature extraction capabilities of these pre-trained networks. To ensure the best results each model was made trainable.

## Training

All models are compiled using the Adam optimiser, categorical cross entropy loss was used, and metrics were set to accuracy. Additionally, each model had a custom metrics callback to record F1 scores over each epoch, providing a more comprehensive evaluation of model performance. The custom model also included a checkpoint callback to save the model to the file system during training.

All models were trained over 20 epochs, and the final created model is saved to the file system.

# Results

## Accuracy

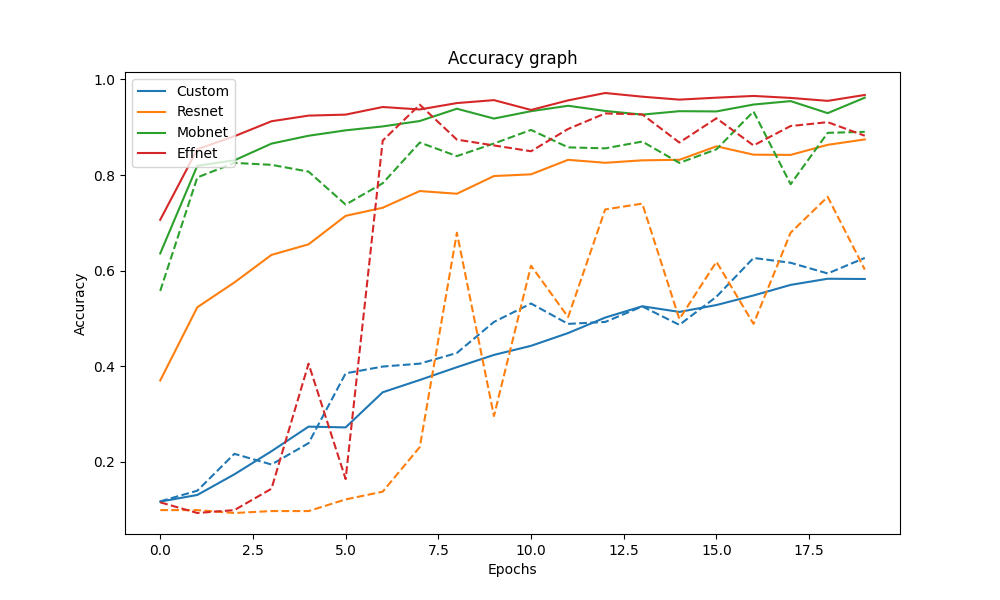


Figure 3 Accuracy graph

Accuracy graph shows that ResNet class greatest overfitting. EffNet starts with a lot of overfitting and then performs reasonably well after 5th epoch. MobNet does not have a lot of overfitting and performs similarly to EffNet. Custom model is the best regarding overfitting. Custom model has the worst score while the Resnet has worst overfitting. The best models are MobNet and EffNet. The accuracy of custom model is increasing thill the last epoch and could benefit from more epoch. MobNet and EffNet do not gain much after 5th epoch with ResNet gaining accuracy thill 15th epoch.

## Loss

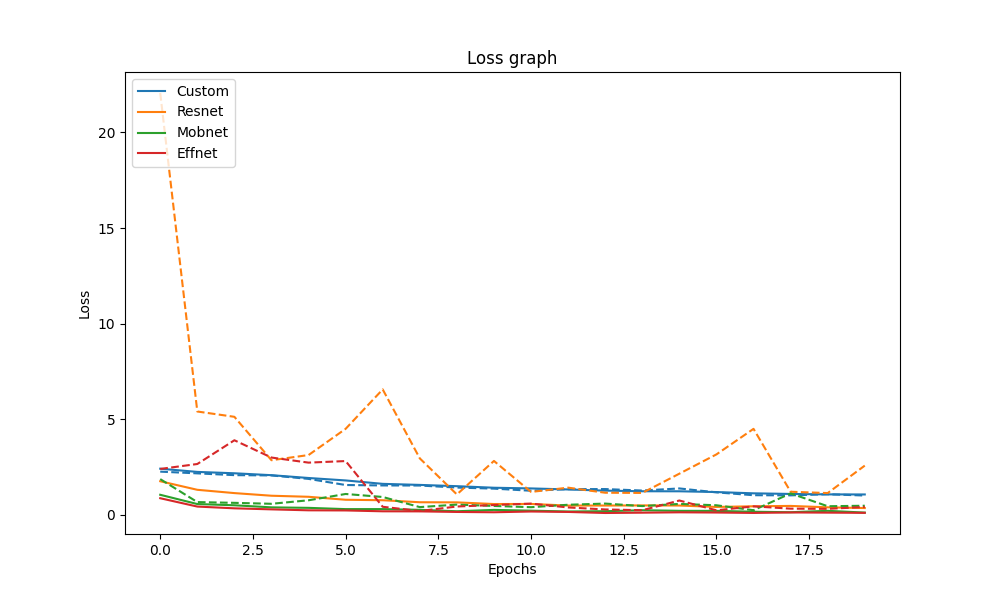


Figure 4 Loss graph

The loss graph values have opposite tendency to the accuracy graph what is desired outcome. With ResNet having a large spike on the 1st epoch with validation data.

## F1 score

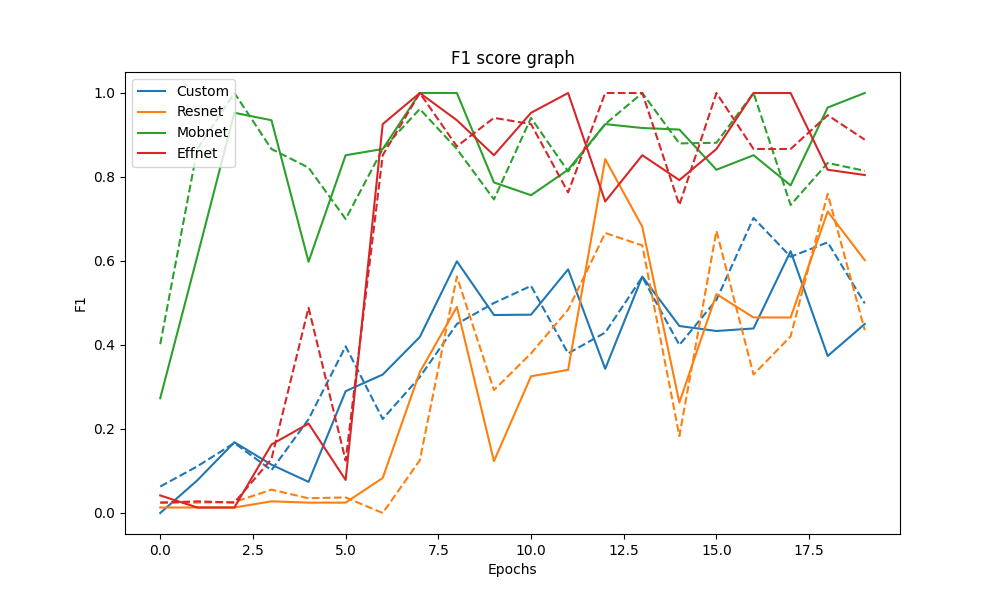


Figure 5 F1 score graph

F1 score graph confirms shows that ResNet model performs on par with custom mode with average score. EffNet results are on par with MobNet from 5th epoch with MobNet having good results from the start.

## Socres summary

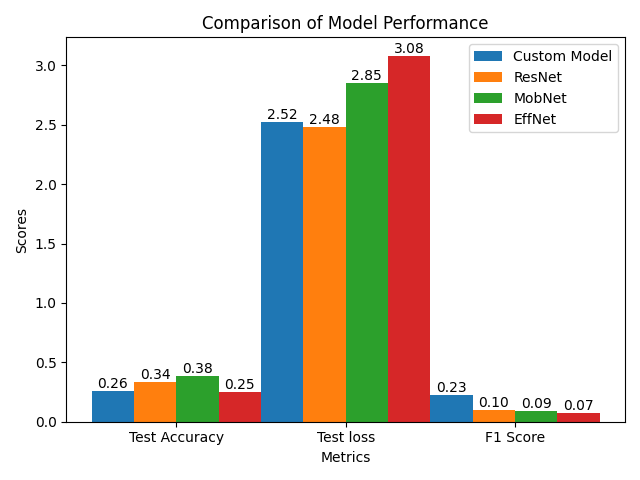


Figure 2 Model scores

Summary of the results on test dataset shows that none of the models are great and that custom model is somewhat of an outlier regarding F1 score. EffNet model performs the worst with highest loss and lowest F1 and accuracy score. MobNet has higher accuracy but lower loss and F1 score with ResNet being opposite to MobNet.

## Confusion matrix

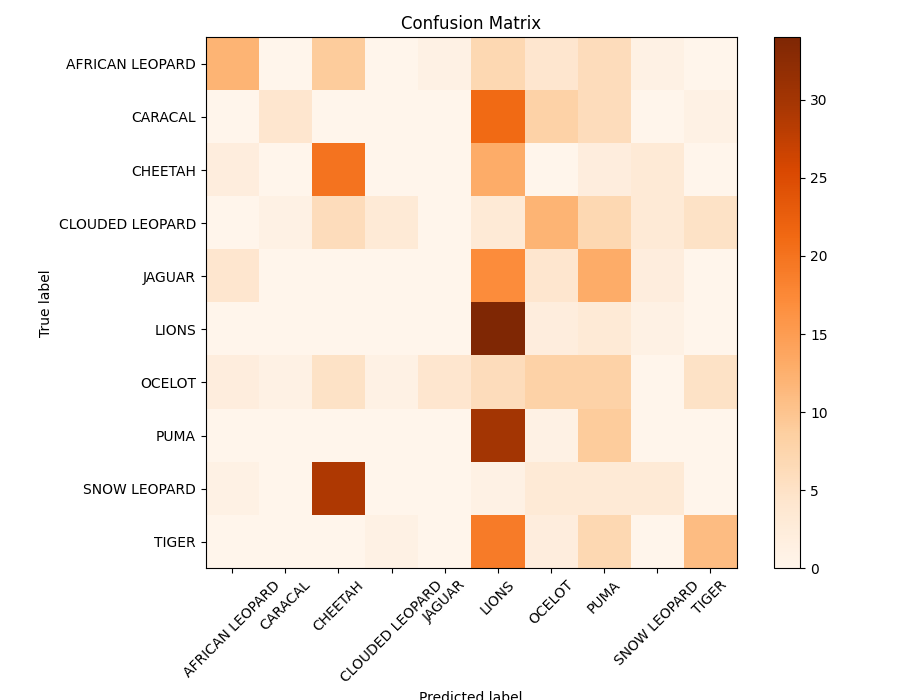


Figure 6 Confusion matrix

Confusion matrix displays the overfitting of the model for lions, lack of detection for snow leopards and jaguars. Other classes are not recognized greatly but they are spread evenly.

## Classification reports

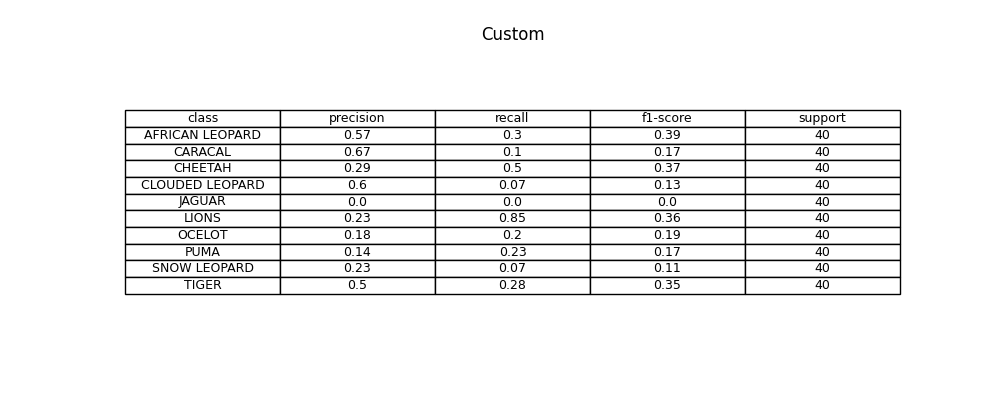


Figure 7 Custom model classification report

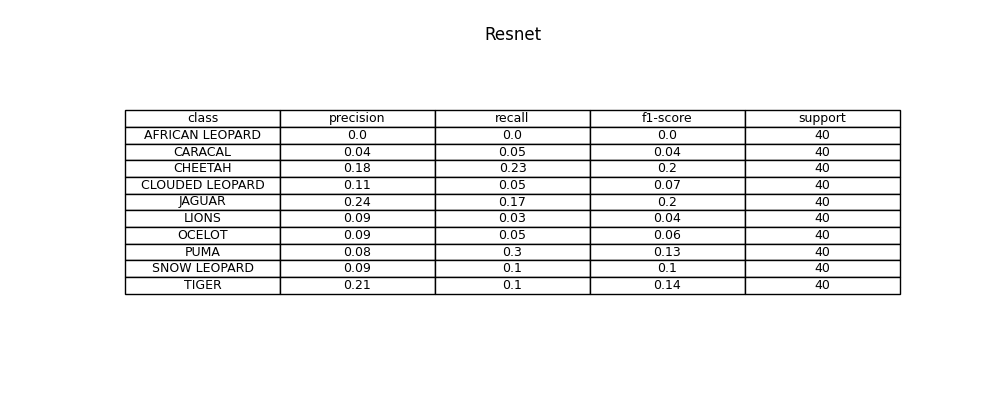


Figure 8 ResNet model classification report

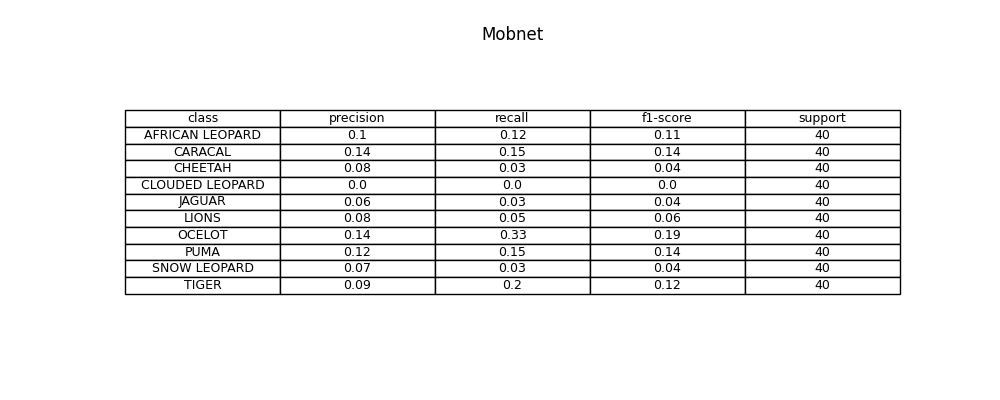


Figure 9 MobNet model classification report

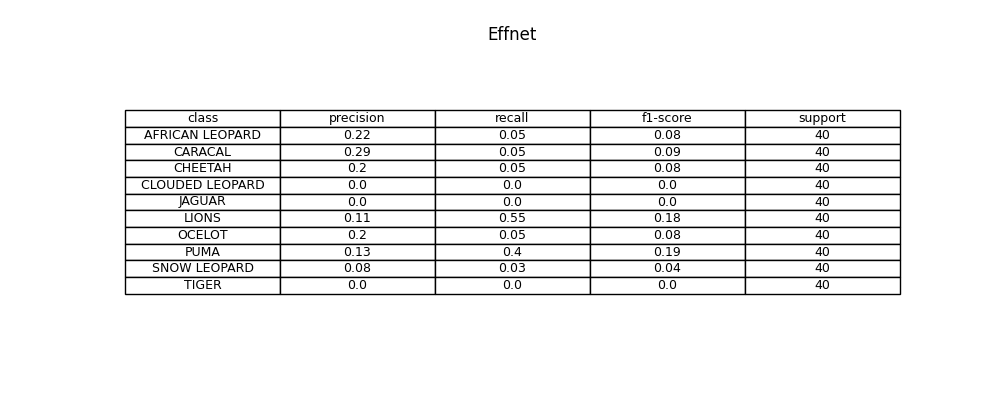


Figure 10 EffNet model classification report

Classification reports confirm graph data. Support column confirms that each class has 40 images. Custom model does not recognize jaguars, ResNet doesn’t recognize African leopards, MobNet doesn’t recognize clouded leopards and EffNet doesn’t recognize clouded leopards, jaguars and tigers. Custom models has best results, while EffNet has the worst.

## Image example

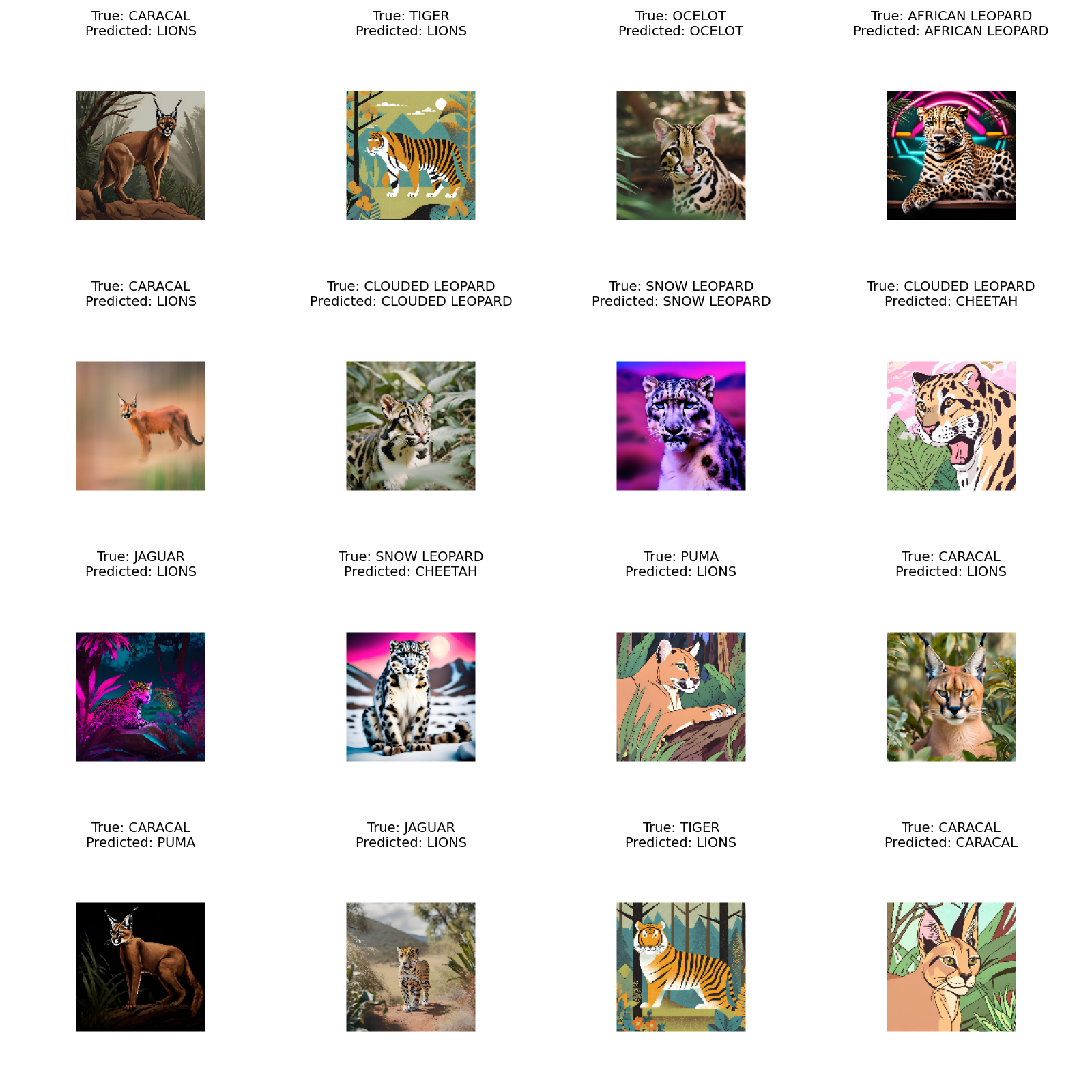


Figure 11 Test prediction visualization

Visualizing the pictures with predicted and true labels it is clear that training scores are better than evaluation scores because AI generated images are not always good representations of real images and models are not trained good enough for classification of such images. This is shown by lions being the most predicted class regardless of the true label.

# Conclusion

Custom model is the best way for solving specific requests such as wild cats detection and classification. If high precision is the main requirement, we need to provide the biggest dataset possible for training validating and testing the model. If it is known that the same kind of pictures will be asked to evaluate after training, it is better to use state-of-the-art models. If the training data is not sufficient enough it is better to create a custom model that will be trained and good enough to use as a solution with additional validation of supervisors.