Image Classification - Cats, Dogs and Butterflies

Amanda Tavares

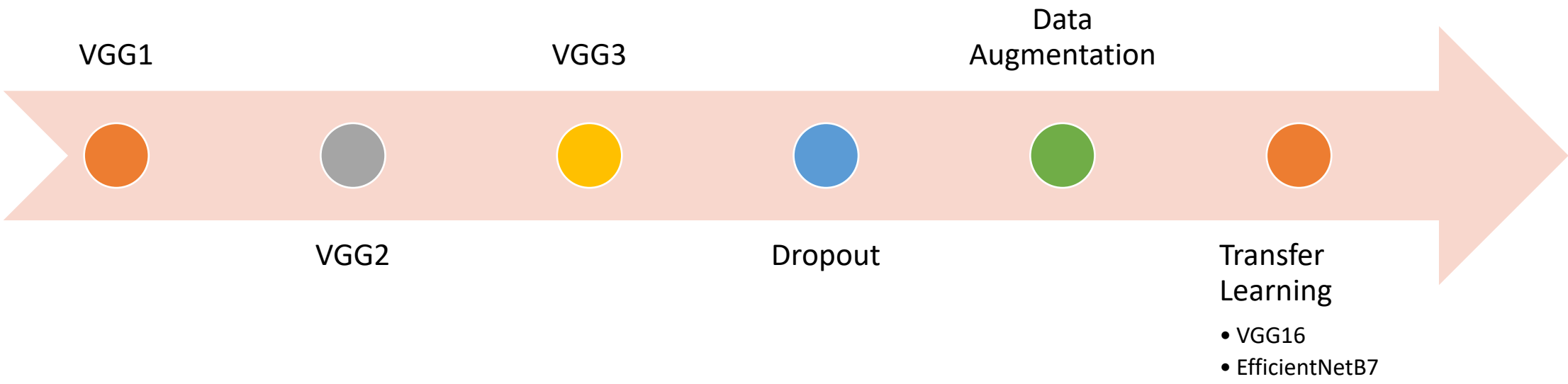
Arina Sanches

Lirielly Nascimento

Outline

- Experiments
- Dataset Analysis
- Data Augmentation
- VGG-16 Architecture
- EficientNet B7 Architecture
- Available pre-trained weights models in Keras
- Execution Results
- Reinforcement Learning
- RL - Execution Results
- Bibliography

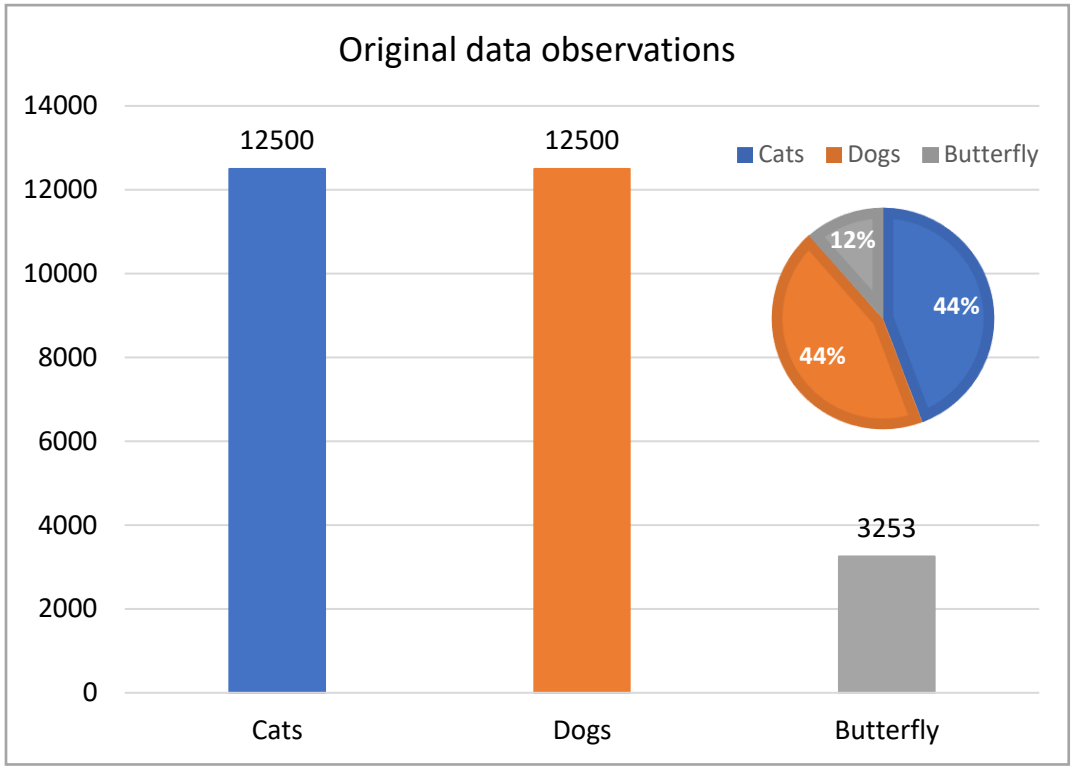
Experiments

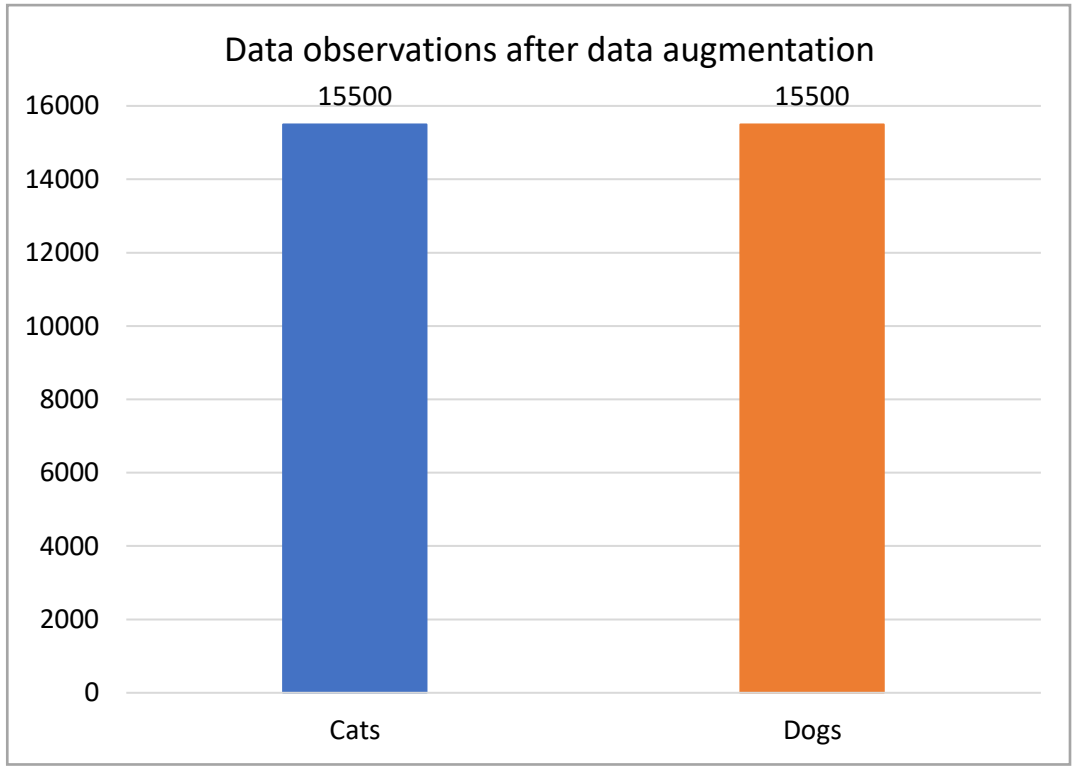


Steps followed

- ✓ 2 classes: dogs and cats
- ✓ 2 classes with data augmentation: dogs and cats
- ✓ 3 classes: dogs, cats and butterfly

Dataset analysis





Data Augmentation

original image

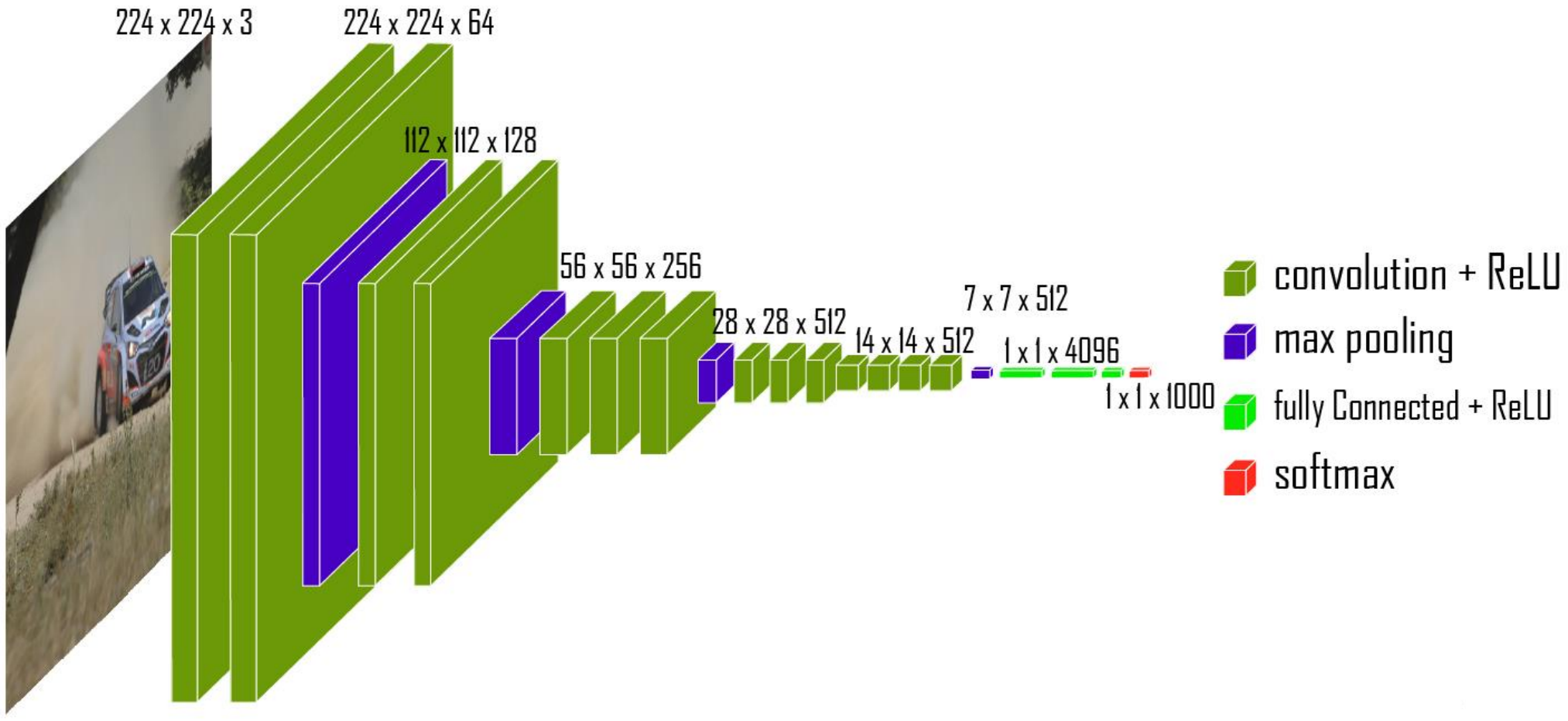
gaussian white noise

+ 1000 for cats
+ 1000 for dogs

blocking mask

+ 2000 for cats
+ 2000 for dogs

VGG-16* Architecture



*depth: 16 weighted layers

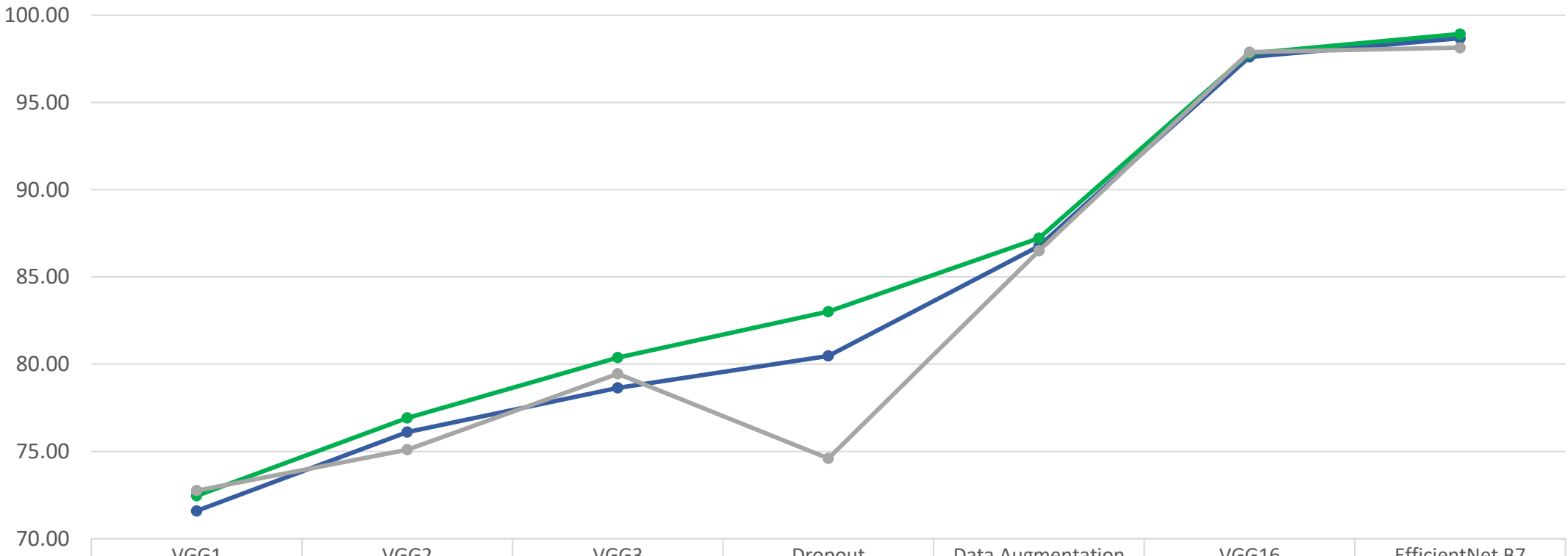
EfficientNet B7* Architecture

*depth: 438 weighted layers

Available pre-trained weights models in Keras

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.00%	94.50%	22.9M	81	109.4	8.1
VGG16	528	71.30%	90.10%	138.4M	16	69.5	4.2
VGG19	549	71.30%	90.00%	143.7M	19	84.8	4.4
ResNet50	98	74.90%	92.10%	25.6M	107	58.2	4.6
ResNet50V2	98	76.00%	93.00%	25.6M	103	45.6	4.4
ResNet101	171	76.40%	92.80%	44.7M	209	89.6	5.2
ResNet101V2	171	77.20%	93.80%	44.7M	205	72.7	5.4
ResNet152	232	76.60%	93.10%	60.4M	311	127.4	6.5
ResNet152V2	232	78.00%	94.20%	60.4M	307	107.5	6.6
InceptionV3	92	77.90%	93.70%	23.9M	189	42.2	6.9
InceptionResNetV2	215	80.30%	95.30%	55.9M	449	130.2	10
MobileNet	16	70.40%	89.50%	4.3M	55	22.6	3.4
MobileNetV2	14	71.30%	90.10%	3.5M	105	25.9	3.8
DenseNet121	33	75.00%	92.30%	8.1M	242	77.1	5.4
DenseNet169	57	76.20%	93.20%	14.3M	338	96.4	6.3
DenseNet201	80	77.30%	93.60%	20.2M	402	127.2	6.7
NASNetMobile	23	74.40%	91.90%	5.3M	389	27	6.7
NASNetLarge	343	82.50%	96.00%	88.9M	533	344.5	20
EfficientNetB0	29	77.10%	93.30%	5.3M	132	46	4.9
EfficientNetB1	31	79.10%	94.40%	7.9M	186	60.2	5.6
EfficientNetB2	36	80.10%	94.90%	9.2M	186	80.8	6.5
EfficientNetB3	48	81.60%	95.70%	12.3M	210	140	8.8
EfficientNetB4	75	82.90%	96.40%	19.5M	258	308.3	15.1
EfficientNetB5	118	83.60%	96.70%	30.6M	312	579.2	25.3
EfficientNetB6	166	84.00%	96.80%	43.3M	360	958.1	40.4
EfficientNetB7	256	84.30%	97.00%	66.7M	438	1578.9	61.6

Execution Results



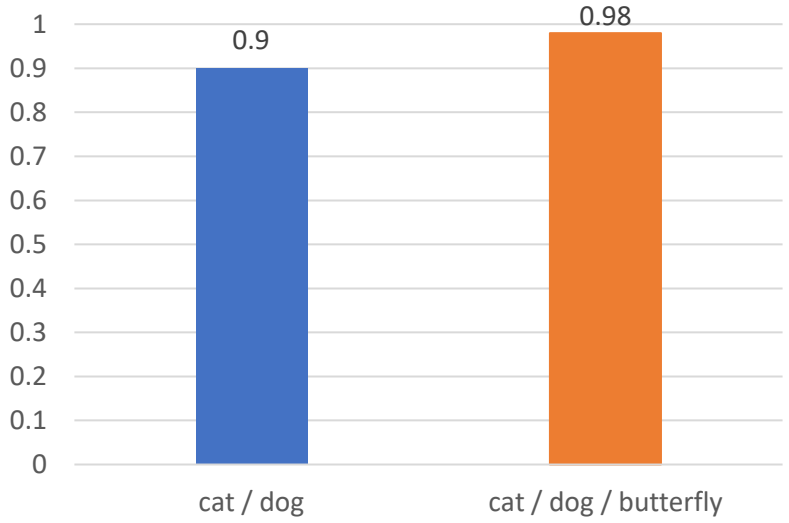
	VGG1	VGG2	VGG3	Dropout	Data Augmentation	VGG16	EfficientNet B7
cat / dog	71.59	76.11	78.64	80.48	86.77	97.61	98.68
cat / dog - data augmentation	72.46	76.92	80.38	83.02	87.22	97.81	98.92
cat / dog / butterfly	72.76	75.10	79.45	74.61	86.50	97.89	98.14

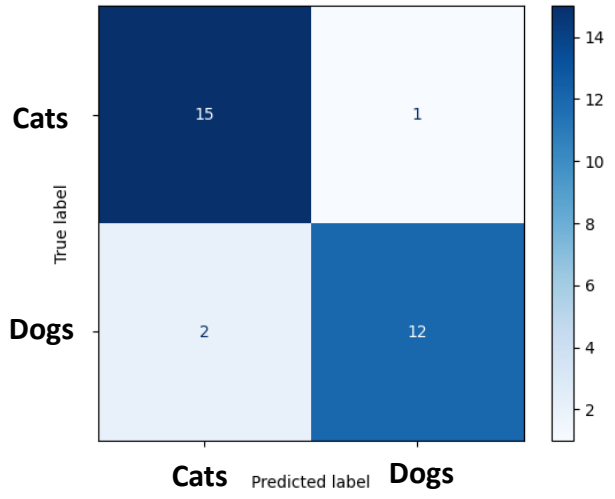
cat / dog cat / dog - data augmentation cat / dog / butterfly

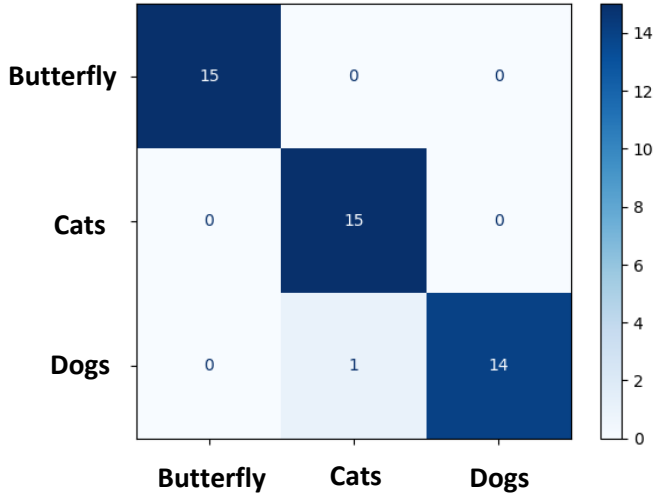
Another way of training using tabular data

- Features created:
  - size of nose;
  - shape of ears;
  - hair;
  - wings;
  - animal size;
  - antenna.

- Dataset observations:
  - 50 observations for each class (cat, dog and butterfly).





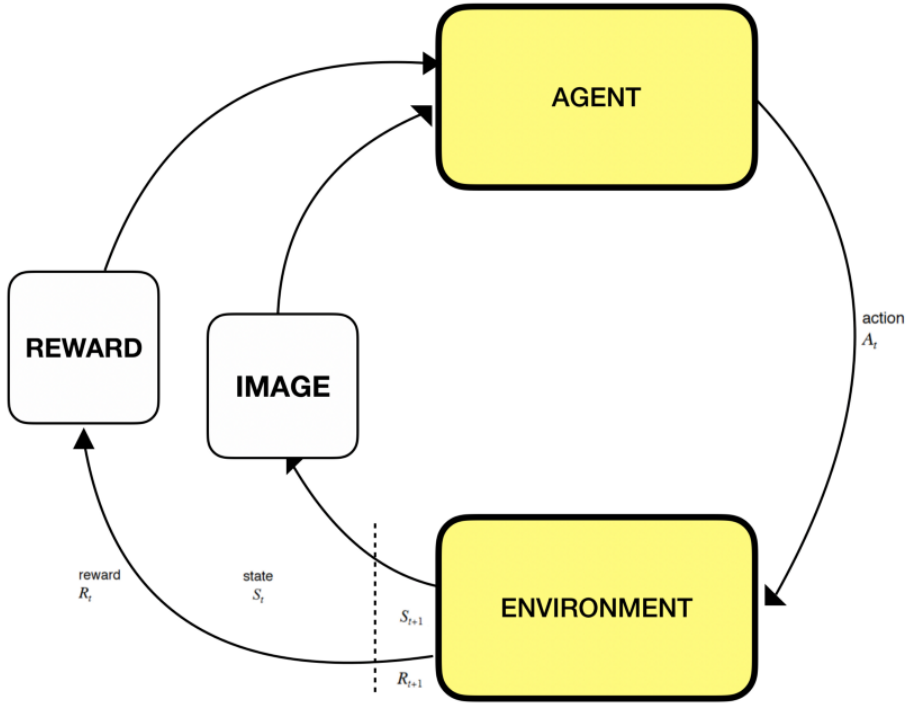


Reinforcement Learning

1 - State: The state ***s*** in the environment is the training sample, which would be the image sample in our case.

2- Action: The action ***a*** of the agent is the label of the training sample. As we have a binary problem, where the agent is only able to choose from the set of actions ***A = {0,1}***, ***0*** is the **cat** class and ***1*** is the **dog** class.

3- Reward: The reward ***r*** is the feedback that the environment gives back to the agent for it to measure its success in classifying the state ***s*** correctly.

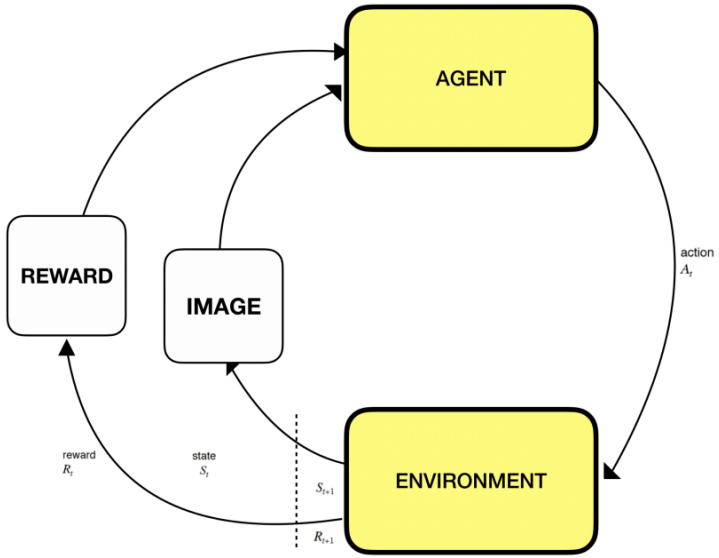


Reinforcement Learning

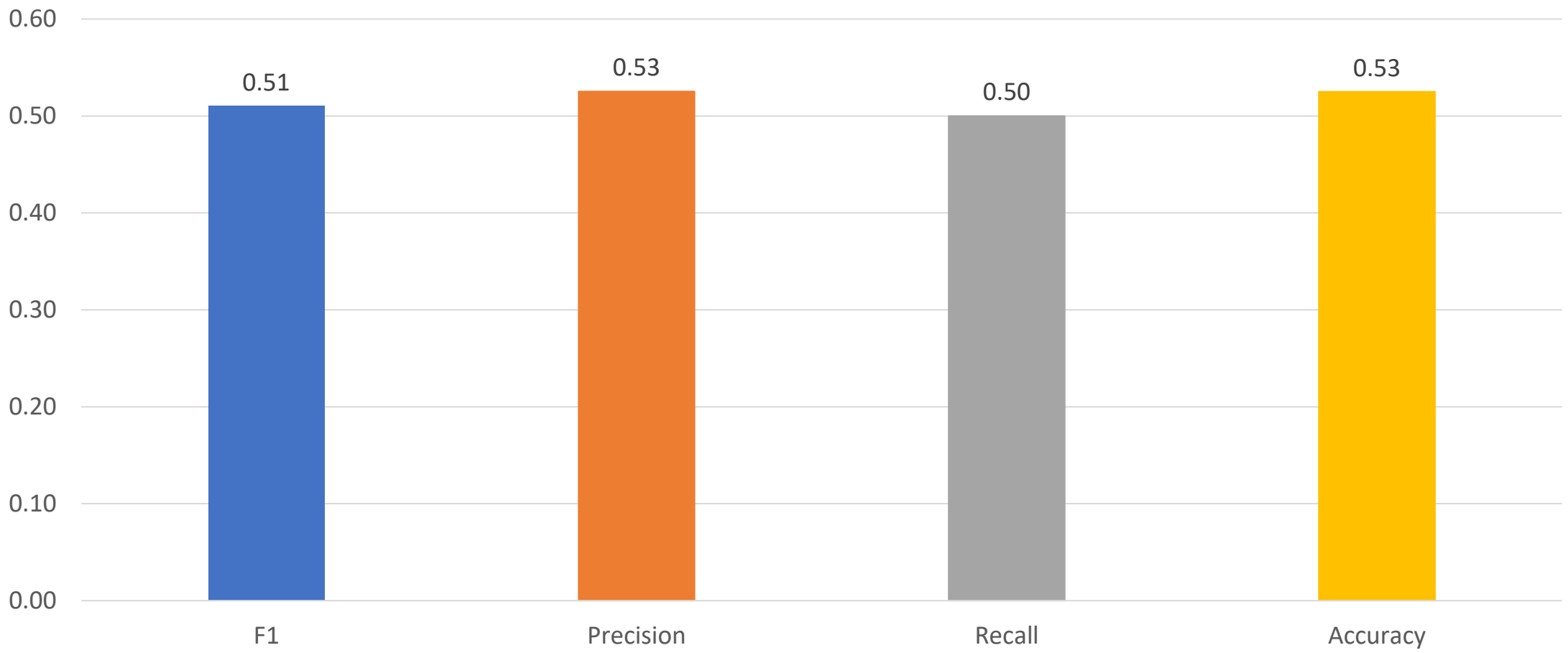
4- Discount factor: The factor $\gamma \in [0,1]$, it weighs in the importance of future rewards. Since we are working on image classification, then consecutive samples are not correlated, and each image needs to be classified correctly. Therefore, a low value for γ would be a better choice.

5- Exploration rate: The rate $\varepsilon \in [0,1]$, when it is set to 1, this means that the actions taken are purely based on exploration, on the other hand, if the value is 0 the actions taken are an exploitation of the agent's knowledge.

6- Episode: The episode ***e*** ends when all the training samples in the training set have been passed for the agent to classify.



RL - Execution Results



Bibliography

- How to Classify Photos of Dogs and Cats
- Keras Applications
- Complete Architectural Details of all EfficientNet Models
- VGG-16 | MODELO CNN
- Using deep q-learning in the classification of an imbalanced dataset