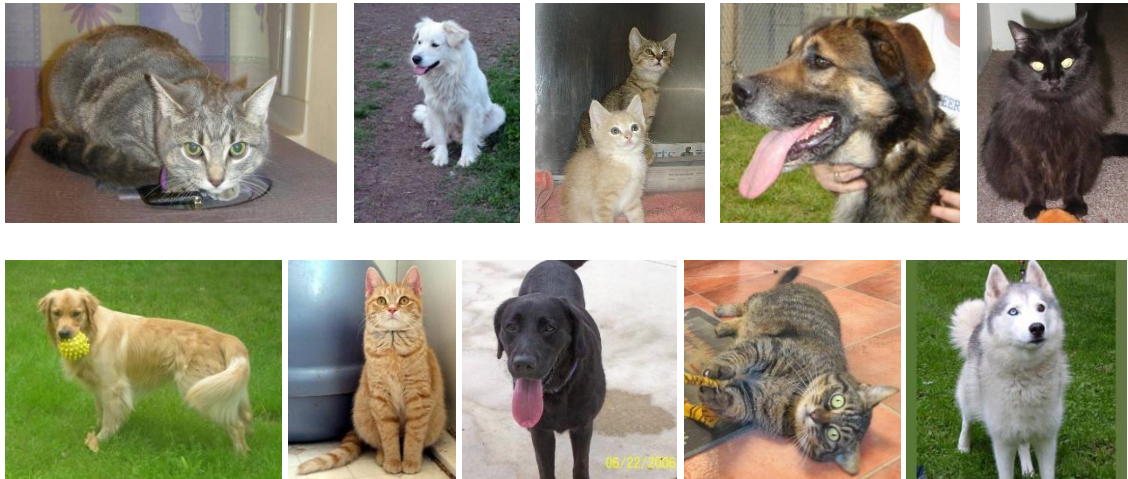


APVC – Desafio 2

Redes neuronais convolucionais para distinguir cães de gatos

Neste desafio pretende-se definir e treinar redes neuronais convolucionais capazes de distinguir entre imagens de cães e imagens de gatos.



CATS AND DOGS

Parte 1 – Preparação dos dados

Descarregue o *dataset Cats and Dogs*¹, disponibilizado no Moodle. Este dataset é composto por 2.000 imagens para treino (1.000 cães e 1.000 gatos); e 1.000 imagens para validação (500 cães e 500 gatos).

Processe o *dataset* de forma a obter novos conjuntos para validação e teste. Para tal, deverá separar as 1.000 imagens do conjunto de validação original segundo dois novos conjuntos, um que será usado para validação e outro para teste, cada um com 500 imagens. Após este processo deverá ficar com 2.000 imagens para treino, 500 para validação e 500 para teste.

Utilize as funções do *Tensorflow/Keras* `keras.utils.image_dataset_from_directory(...)`² e `keras.utils.split_dataset(...)`³ para facilitar a construção dos *datasets*. Note também que neste problema de classificação poderá utilizar uma abordagem de classificação binária, o que poderá ter algumas implicações na parametrização da construção dos *datasets* a partir do sistema de ficheiros.

Verifique a distribuição das classes nos *datasets* de validação e teste obtidos. Cada um deveria idealmente conter 250 imagens de cães e 250 imagens de gatos. Dado o carácter pseudo-aleatório dos processos envolvidos, poderá não chegar a esta distribuição ideal, mas certifique-se que a distribuição de classes não fica muito distante do ideal.

¹ É um subconjunto das imagens publicadas em: <https://www.kaggle.com/c/dogs-vs-cats/data>

² https://keras.io/api/data_loading/image/

³ https://keras.io/api/utils/python_utils/#splitdataset-function

Parte 2 – Rede neuronal convolucional “custom”

Defina uma rede neuronal convolucional que lhe permita classificar as imagens em *Cão* ou *Gato*. A arquitetura da rede proposta, a dimensão das imagens para entrada na CNN, a abordagem seguida (classificação binária ou multiclasse), a utilização de técnicas que visem reduzir o *overfitting*, assim como outras parametrizações, ficam totalmente ao seu critério. Tal como no desafio anterior, sugere-se a utilização de *callbacks* para garantir que consegue obter e restaurar os melhores modelos obtidos durante os processos de treino.

Parte 3 – Transferência de conhecimento

Utilize transferência de conhecimento com vista a melhorar os resultados atingidos na Parte 2. Para tal deverá utilizar como base duas CNNs pré-treinadas à sua escolha, de entre as disponibilizadas pelo *Tensorflow/Keras*⁴. A única restrição é que as duas redes deverão ser provenientes de duas famílias distintas. Veja a documentação para determinar se, consoante as arquiteturas escolhidas, será ou não necessário aplicar algum tipo de pre-processamento às imagens (por exemplo normalizar os valores dos pixels, redimensionar as imagens, trocar canais de cor, etc.).

Parte 4 – “Mini-Relatório”

Escreva um “mini-relatório” para descrever sucintamente o que foi feito, assim como para mostrar e comentar os resultados relativos às partes 1, 2 e 3 do desafio. Deverá também incluir no relatório:

- exemplos de imagens do conjunto de teste em que o seu melhor modelo falha a classificação;
- o resultado da classificação em fotografias de cães e gatos dos membros do grupo. Caso ninguém no grupo tenha um cão ou um gato como animal de estimação, poderá pedir a outra pessoa ou procurar na internet (o que interessa é que sejam imagens que não estejam presentes no *dataset*).

O mini-relatório deverá ter cerca de 5 páginas (sem contar com capa e eventuais anexos).

Entrega e avaliação

A entrega do desafio é realizada através do moodle e basta que um dos elementos do grupo submeta o trabalho, entregando os ***scripts (ou notebooks) desenvolvidos***, um pdf com o “**mini-relatório**” e eventuais ficheiros com os pesos das ligações das redes treinadas. O deadline da entrega é o **dia 30/março (domingo) às 23:59**.

A nota será dada numa escala de 0 a 10 e será dado feedback através do moodle. A parte 1 vale 10% da nota do desafio e as partes 2, 3 e 4 valem 30% cada uma. Trabalhos entregues após o deadline são penalizados com um ponto por cada dia em atraso.

Em caso de dúvida esteja à vontade para contactar o professor, tanto nas aulas como por email.

⁴ https://www.tensorflow.org/api_docs/python/tf/keras/applications