

1. Estrutura e estatísticas do novo dataset

Exportado de

<https://universe.roboflow.com/car-damage-detection-cardd/car-damage-severity-detection-cardd>

em YOLOv8, o dataset contém 3 pastas: 'test', 'train', 'valid', e um ficheiro config:'data.yaml' .
com split 70/20/10:

Split	# imagens	# labels
Train	1 395	1 395
Val	401	401
Test	204	204

Contrariamente ao dataset anterior, neste as classes são definidas no **data.yaml** (já não há sub-pastas por classe).

Exemplo de imagem com respectivas labels:



- Fizemos uma verificação de hashes e não encontramos duplicados entre train e val.
- Fomos também perceber como estavam distribuídas as classes pelos diferentes sets :

split	car-part-crack	detachment	flat-tire	glass-crack	lamp-crack	minor-deformation	moderate-deformation	paint-chips	scratches	severe-deformation	side-mirror-crack
train	348	54	119	253	249	246	630	266	991	48	8
valid	73	10	33	70	70	74	174	72	313	13	4
test	50	4	15	35	34	26	86	31	157	8	2

Dúvida 1 – “class imbalance”

Os counts por classe variam muito, algumas classes com 4 e outras com 313. Devemos considerar isto “unbalanced”, ou bastará reforçar com data augmentation?

2. Data augmentation

Tentámos replicar o bloco Keras que usávamos:

```
data_augmentation = keras.Sequential([
    layers.RandomFlip("horizontal",
        input_shape=(img_height,
                      img_width,
                      3)),
    layers.RandomRotation(0.1),
    layers.RandomZoom(0.1),
])
```

Mas pensamos que, como o dataset agora vem em formato YOLO (imagens + ficheiros .txt com bounding boxes), esta estratégia não esteja a ter em conta as labels.

Dúvida 2

Qual seria a forma correcta de fazer augmentation mantendo as boxes e labels corretamente? (Albumentations? Transforms do Ultralytics?)

Se é que é sequer realmente necessário aplicar data augmentation?

3. Variabilidade nos resultados

Rodámos exactamente o mesmo código em dois portáteis diferentes (mesmo número de epochs, mesmo dataset) e obtivemos métricas muito divergentes – por exemplo, para 15 epochs um colega teve 0.52 val-accuracy e outro 0.14. O mesmo portátil a correr em momentos distintos também deu valores diferentes.

Dúvida 3

Isto é expectável ou teremos algum erro de configuração?

Fará sentido fixar resultados, e se recomenda alguma forma (seeds, `tf.keras.utils.set_random_seed`, ou outro...)?

E gostaríamos de perceber se os valores de accuracy e loss que nos aparece nos graficos abaixo são normais?

Resultado do treino, dependendo da escolha das epochs, cada um dos elementos do grupo (cada um representado por imagens abaixo por a e b)



