# Material

Neste conjunto de códigos, foi apenas utilizado a propriedade Material para treinar o modelo. Existem 3 classes para esta propriedade e utilizei a pasta "Pasta Final TFRecord – Material" nos modelos, uma vez que esta contém os ficheiros TFRecord. Abaixo, pode-se observar como os dados estão separados (treino, validação e teste) para cada classe. Esta tabela encontra-se presente no ficheiro "Classes.xlsx".

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Classe** | **Material** | **Quantidade** | **Treino** | **Validação** | **Teste** |
| 0 | Alvenaria | 186 | 146 | 23 | 17 |
| 1 | Aço | 143 | 114 | 13 | 16 |
| 2 | Betão Armado | 6 130 | 4 907 | 609 | 614 |
|  | Total | 6 459 | 5 167 | 645 | 647 |

## ResNet

### Tentativa 1

### Tentativa 2

Caminho no GitHub para o código: Material -> ResNet -> Tentativa 2 -> Material (ResNet) – Tentativa 2.py

Caminho no GitHub para os resultados: Material -> ResNet -> Tentativa 2 e de forma resumida Resultados -> Material -> ResNet -> Tentativa 2.xlsx

**Website utilizado:** <https://keras.io/guides/transfer_learning/>

**O modelo está pré-treinado?** Sim, o modelo ResNet50 é pré-treinado com pesos do ImageNet.

**Tem otimização de hiper parâmetros?** Não, não há otimização de hiperparâmetros implementada.

**Qual é o número de epochs?** 10 epochs.

**Tem regra de paragem antecipada?** Não, não há regra de paragem antecipada.

O resumo do código é o seguinte:

* É carregado o modelo ResNet50 sem a camada final de classificação (include\_top=False), usando pesos do ImageNet.
* São congeladas as camadas do modelo ResNet50 para que estas não sejam treinadas, mantendo as características do ImageNet.
* Depois é construído um novo modelo na parte superior, onde a entrada é definida o tamanho 224x224 e 3 canais de cor. E a camada de saída é uma camada densa com 3 classes e ativação softmax.
* As últimas 10 camadas do modelo ResNet50 são descongeladas para permitir o ajusto durante o treino.
* O modelo é compilado com o otimizador Adam, é usado uma função de perda SparseCategoricalCrossentropy e a métrica accuracy é utilizada para medir o desempenho.

# Material + Ano

Neste conjunto de códigos, foi apenas utilizado a propriedade Material e Ano para treinar o modelo. Existem 5 classes para esta propriedade e utilizei a pasta "Pasta Final TFRecord – Material + Ano" nos modelos, uma vez que esta contém os ficheiros TFRecord. Abaixo, pode-se observar como os dados estão separados (treino, validação e teste) para cada classe. Esta tabela encontra-se presente no ficheiro "Classes.xlsx".

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Classe** | **Material** | **Ano** | **Qtd** | **Treino** | **Validação** | **Teste** |
| 0 | Alvenaria |  | 186 | 155 | 19 | 12 |
| 1 | Aço | Antes de 1983 | 64 | 54 | 5 | 5 |
| 2 | Aço | Depois de 1983 | 79 | 62 | 7 | 10 |
| 3 | Betão Armado | Antes de 1983 | 579 | 464 | 58 | 57 |
| 4 | Betão Armado | Depois de 1983 | 5 551 | 4 432 | 556 | 563 |
|  |  | Total | 6459 | 5167 | 645 | 647 |

## ResNet

### Tentativa 1

### Tentativa 2

#### Do Zero

Caminho no GitHub para o código: Material + Ano -> ResNet -> Tentativa 1 -> Do Zero -> Material + Ano (ResNet) - Tentativa 1 Do Zero.py

Caminho no GitHub para os resultados: Material + Ano -> ResNet -> Tentativa 1 -> Do Zero 1 e de forma resumida Material + Ano -> ResNet -> Tentativa 1 -> Do Zero.xlsx

**Website utilizado:** <https://keras.io/guides/transfer_learning/>

**O modelo está pré-treinado?** Sim, o modelo ResNet50 é pré-treinado com pesos do ImageNet.

**Tem otimização de hiper parâmetros?** Não, não há otimização de hiperparâmetros implementada.

**Qual é o número de epochs?** 10 epochs.

**Tem regra de paragem antecipada?** Não, não há regra de paragem antecipada.

O resumo do código é o seguinte:

* É carregado o modelo ResNet50 sem a camada final de classificação (include\_top=False), usando pesos do ImageNet.
* São congeladas as camadas do modelo ResNet50 para que estas não sejam treinadas, mantendo as características do ImageNet.
* Depois é construído um novo modelo na parte superior, onde a entrada é definida o tamanho 224x224 e 3 canais de cor. E a camada de saída é uma camada densa com 3 classes e ativação softmax.
* As últimas 10 camadas do modelo ResNet50 são descongeladas para permitir o ajusto durante o treino.
* O modelo é compilado com o otimizador Adam, é usado uma função de perda SparseCategoricalCrossentropy e a métrica accuracy é utilizada para medir o desempenho.

# Material + Ano + Estrutura