

Wybrałem model ResNet50 z kilkoma modyfikacjami, ponieważ jest to jeden z najskuteczniejszych i najpopularniejszych modeli konwolucyjnych sieci neuronowych (CNN) do rozpoznawania obrazów, a jego architektura dobrze nadaje się do rozpoznawania ras psów. Oto kilka powodów, dlaczego wybrałem właśnie ResNet50 oraz podejście z transfer learning:

1. Skuteczność ResNet50

ResNet50 to głęboka sieć CNN z 50 warstwami, która została zaprojektowana specjalnie z myślą o rozwiązywaniu problemów zanikającego gradientu (vanishing gradient problem) dzięki wykorzystaniu tzw. "residual connections". Te połączenia resztkowe umożliwiają trenowanie bardzo głębokich sieci bez utraty informacji i sprawiają, że model jest bardziej wydajny w rozpoznawaniu obrazów, zwłaszcza gdy klasy mają drobne różnice, jak to jest w przypadku ras psów.

2. Transfer Learning

Korzystając z modelu ResNet50 wstępnie wytrenowanego na zestawie danych ImageNet, możemy skorzystać z gotowych, wytrenowanych filtrów, które są już wyspecjalizowane w wykrywaniu podstawowych cech obrazów (krawędzie, kolory, tekstury itp.). Dzięki transfer learning nie musimy trenować całej sieci od zera, co pozwala:

Skrócić czas trenowania – model już „wie”, jak rozpoznawać podstawowe wzorce, co przyspiesza trenowanie.

Zmniejszyć wymagania na dane – transfer learning jest szczególnie pomocny, gdy mamy ograniczoną liczbę obrazów treningowych, ponieważ model wstępnie wytrenowany na dużym zestawie (ImageNet) już ma „wiedzę”, którą możemy dostosować do konkretnego zadania.

3. Generalizacja

ResNet50, dzięki swojej architekturze i połączeniom resztkowym, wykazuje dużą zdolność do generalizacji. Oznacza to, że dobrze radzi sobie z nowymi, nieznanymi obrazami, co jest szczególnie ważne przy rozpoznawaniu ras psów, gdzie psy tej samej rasy mogą różnić się wyglądem.

4. Obsługa złożoności i podobieństwa między rasami

Rozpoznawanie ras psów jest trudnym zadaniem, ponieważ niektóre rasy są bardzo do siebie podobne, a różnice mogą być subtelne. Głębokie sieci, takie jak ResNet50, są w stanie zidentyfikować te subtelne różnice dzięki dużej liczbie warstw, które wyodrębniają szczegółowe cechy.

5. Elastyczność – możliwość dalszego dostrajania (fine-tuning)

Architektura ResNet50 pozwala na "fine-tuning", czyli dalsze trenowanie na bardziej zaawansowanych warstwach modelu, co umożliwia lepsze dopasowanie modelu do konkretnego zbioru danych. Możemy zablokować wcześniejsze warstwy, aby zachować podstawowe cechy wykrywania krawędzi i tekstur, a odblokować bardziej zaawansowane warstwy, aby model mógł skupić się na specyficznych cechach charakterystycznych dla ras psów.

O to logi oraz wyniki z pracy zaimplementowanego prototypowego modelu:

Epoch 1/10

515/515 ————— 1512s 3s/step - accuracy: 0.4262 - loss:
2.4229 - val_accuracy: 0.7119 - val_loss: 0.9270

Epoch 2/10

515/515 ————— 1206s 2s/step - accuracy: 0.6858 - loss:
1.0111 - val_accuracy: 0.7386 - val_loss: 0.8343

Epoch 3/10

515/515 ————— 1113s 2s/step - accuracy: 0.7370 - loss:
0.8427 - val_accuracy: 0.7522 - val_loss: 0.8192

Epoch 4/10

416/515 ————— 3:12 2s/step - accuracy: 0.7576 - loss:
0.7647

515/515 ————— 1403s 3s/step - accuracy: 0.7572 - loss:
0.7680 - val_accuracy: 0.7478 - val_loss: 0.8544

Epoch 5/10

426/515 ————— 2:21 2s/step - accuracy: 0.7745 - loss:
0.7290

515/515 ————— 1106s 2s/step - accuracy: 0.7731 - loss:
0.7307 - val_accuracy: 0.7653 - val_loss: 0.8295

Epoch 6/10

515/515 ————— 0s 2s/step - accuracy: 0.7824 - loss: 0.6772

515/515 ————— 1077s 2s/step - accuracy: 0.7824 - loss:
0.6773 - val_accuracy: 0.7551 - val_loss: 0.8202

Epoch 7/10

515/515 ————— 1010s 2s/step - accuracy: 0.7985 - loss:
0.6336 - val_accuracy: 0.7617 - val_loss: 0.8043

Epoch 8/10

515/515 ————— 0s 2s/step - accuracy: 0.8071 - loss: 0.5818

515/515 ————— 1087s 2s/step - accuracy: 0.8071 - loss:
0.5819 - val_accuracy: 0.7668 - val_loss: 0.8421

Epoch 9/10

515/515 ————— 1071s 2s/step - accuracy: 0.8222 - loss:
0.5618 - val_accuracy: 0.7643 - val_loss: 0.8649

Epoch 10/10

515/515 ————— 1140s 2s/step - accuracy: 0.8192 - loss:
0.5468 - val_accuracy: 0.7680 - val_loss: 0.8341

129/129 ————— 171s 1s/step - accuracy: 0.7714 - loss:
0.8128

Validation Accuracy: 0.77

Process finished with exit code 0