



MEMORIA DE PRÁCTICAS: CAR MODEL IDENTIFICATION

Visión por Computador

Enric Bonet Cortés
Máster en IARFID - Universidad Politécnica de Valencia

Car Model Identification

Para este ejercicio, se cuenta con un dataset de 1500 imágenes de coches correspondientes a 20 marcas diferentes de coches, de las cuales 791 pertenecen al corpus de train y 784 al de test.

El mayor reto que supone este ejercicio, es que este problema no se puede resolver con una topología simple, sino que, para superarlo, se debe de implementar un modelo bilinear como el visto en la asignatura, procurando obtener la mayor precisión posible.

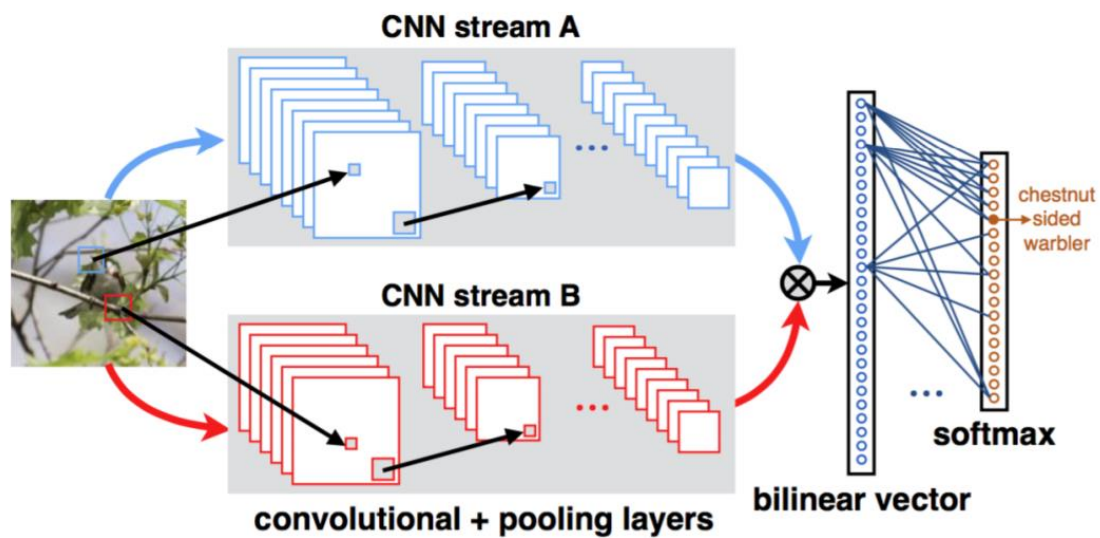


Figura 1: Ejemplo de red Bi-Linear

Se parte en esta ocasión de un fichero con una estructura base que permite cargar el dataset de Dropbox y lo separa en los diferentes corpus, almacenando también sus diferentes clases.

Una vez asimilada la funcionalidad de este Notebook base, se procede a cargar la **VGG16** que proporciona la librería Keras, cuyo modelo ya está pre-entrenado. Una vez cargada dicha red convolucional, se utiliza un bucle para recorrer cada una de sus capas y asignarle a su atributo **trainable** un valor false, lo cual permite congelar las capas de forma que no se reentrenen cuando se vayan a utilizar.

Tras esto, se conecta la última capa de salida (antes de las capas "Fully Connected", la cual se llama "**block5_pool**") del modelo pre-entrenado a dos capas diferentes de Dropout con factor 0.5. Esto permite que se simule tener dos modelos diferentes, pues el Dropout activará por cada parte solo el 50% de las salidas del modelo, para después aplicar la función **outer_product** (función que multiplica las salidas de las dos redes y que ha sido proporcionada por el fichero Notebook base) y aplicarle una última capa densa '**softmax**' ("Full Connected") de 20 neuronas correspondientes a las 20 clases.

Una vez definido dicho modelo, lo único que se puede modificar para lograr un nivel mayor de acierto, es la capa de salida del modelo pre-entrenado que se pasar a las capas de Dropout o el Learning Rate.

En cuanto al Learning Rate, para esta ocasión se ha optado por utilizar un optimizador diferente (en concreto el Adam), que junto a un esquema de reducción del factor de aprendizaje como el que se muestra en la siguiente imagen, han podido obtener mejores resultados respecto al modelo inicialmente descrito.

Por otra parte, para capa de salida de la VGG16 pre-entrenada que se conecta con las capas de Dropout, se ha probado con 8 diferentes: **block5_pool**, **block5_conv3**, **block5_conv2**, **block5_conv1**, **block4_pool**, **block4_conv3**, **block4_conv2** y **block4_conv1**. De estas 8 capas, la que mejores resultados ha proporcionado es la de **block4_conv3**, obteniéndose un acierto superior al 65%.

```
25/24 [=====] - 19s 759ms/step - loss: 0.1170 - accuracy: 0.9899 - val_loss: 1.0743 - val_accuracy: 0.6633
Epoch 140/150
25/24 [=====] - 19s 754ms/step - loss: 0.1194 - accuracy: 0.9949 - val_loss: 1.1013 - val_accuracy: 0.6531
Epoch 141/150
25/24 [=====] - 19s 766ms/step - loss: 0.1192 - accuracy: 0.9949 - val_loss: 1.0984 - val_accuracy: 0.6543
Epoch 142/150
25/24 [=====] - 19s 765ms/step - loss: 0.1167 - accuracy: 0.9949 - val_loss: 1.0976 - val_accuracy: 0.6531
Epoch 143/150
25/24 [=====] - 19s 754ms/step - loss: 0.1185 - accuracy: 0.9937 - val_loss: 1.0881 - val_accuracy: 0.6556
Epoch 144/150
25/24 [=====] - 19s 749ms/step - loss: 0.1200 - accuracy: 0.9924 - val_loss: 1.0984 - val_accuracy: 0.6569
Epoch 145/150
25/24 [=====] - 19s 757ms/step - loss: 0.1106 - accuracy: 0.9949 - val_loss: 1.0974 - val_accuracy: 0.6569
Epoch 146/150
25/24 [=====] - 19s 755ms/step - loss: 0.1191 - accuracy: 0.9937 - val_loss: 1.0941 - val_accuracy: 0.6543
Epoch 147/150
25/24 [=====] - 19s 749ms/step - loss: 0.1138 - accuracy: 0.9912 - val_loss: 1.1023 - val_accuracy: 0.6518
Epoch 148/150
25/24 [=====] - 19s 753ms/step - loss: 0.1106 - accuracy: 0.9937 - val_loss: 1.1050 - val_accuracy: 0.6467
Epoch 149/150
25/24 [=====] - 19s 754ms/step - loss: 0.1190 - accuracy: 0.9899 - val_loss: 1.0962 - val_accuracy: 0.6480
Epoch 150/150
25/24 [=====] - 19s 759ms/step - loss: 0.1134 - accuracy: 0.9937 - val_loss: 1.0973 - val_accuracy: 0.6543
```

Figura 2: Resultados de Car Model