



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Máster en Inteligencia Artificial, Reconocimiento de Formas e Imagen Digital  
Universitat Politècnica de València

## Identificación de modelos de coche

Visión por Computador

*Autor:* Juan Antonio López Ramírez

Curso 2019-2020



## Introducción

En este trabajo, se va a explicar el procedimiento seguido para la correcta realización de un modelo que pueda diferenciar entre 20 modelos distintos de coches.

Para ello, se han implementado redes convolucionales bi-lineales, cuya estructura consiste en dos topologías convolucionales independientes. Las salidas de ambas son combinadas en un vector bilineal. La estructura típica de este modelo se puede apreciar en la figura 1.

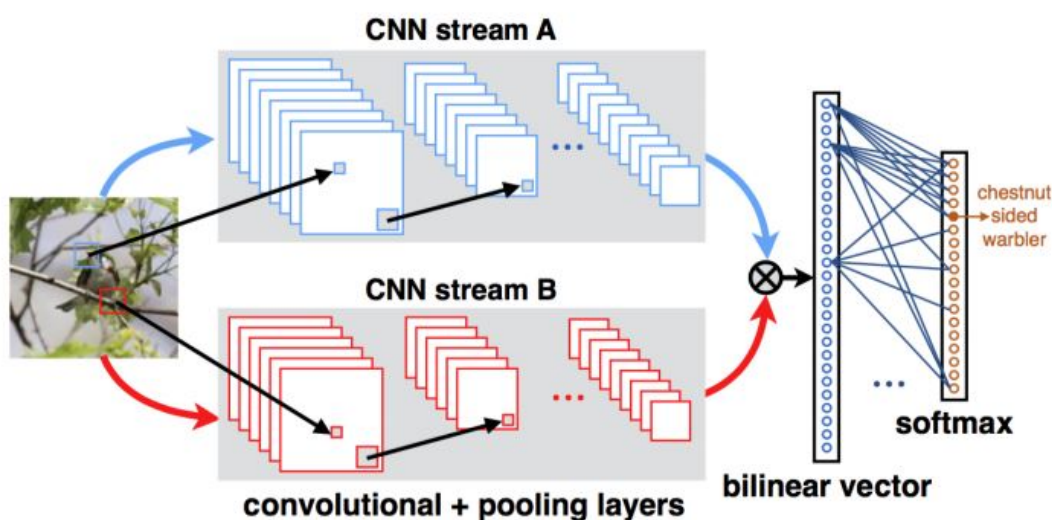


Figura 1: Forma estándar de un modelo bi-lineal.

El dataset empleado consta de 1575 imágenes de 250x250 píxeles en formato RGB.

## Modelo 1. Bilineal con diferentes CNN

El primer modelo que se ha realizado ha consistido en dos redes distintas en cada uno de los flujos del modelo bilineal.

Partiendo del código que se ha proporcionado<sup>1</sup>, se han realizado cuatro experimentos, variando las *epochs* en entrenamiento, los bloques convolucionales empleados, el optimizador y el *learning rate* inicial.

Además, se ha sustituido el *Learning rate scheduler* implementado manualmente por un método de Keras denominado *ReduceLROnPlateau*, donde se modifica el factor de aprendizaje en función de si una métrica mejora sus resultados después de determinadas *epochs*.

Ya con el código completamente funcional se han lanzado los cuatro experimentos, que han arrojado los siguientes resultados:

Como se puede observar, el mejor resultado se obtiene una vez hemos pasado de utilizar el optimizador SGD al Adam, obteniendo un **32.91 %** de precisión.

<sup>1</sup><https://github.com/RParedesPalacios/ComputerVisionLab/blob/master/src/cars1.py>

Epochs	Conv Blocks (En cada CNN)	Optimizador	lr inicial	Resultados (%)
150	(32,64,128)	SGD	0.1	6.89
150	(16,32,64)	SGD	0.1	7.02
500	(16,32,64)	SGD	0.01	7.02
500	(16,32,64)	Adam	0.001	<b>32.91</b>

**Tabla 1:** Resultados del modelo bilineal con diferentes CNN.

## Modelo 2. Bilineal con dos VGG16 preentrenadas

En este segundo modelo, en el que se han sustituido los bloques convolucionales por dos VGG16 preentrenadas con *Imagenet*, se han tenido cuenta tres casos.

Por un lado, se ha entrenado este modelo bilineal entrenando ambas VGG16 sin congelar los pesos.

Por otro lado, se ha probado a entrenar congelando los pesos de una de las dos VGG.

Finalmente, el último entrenamiento ha consistido en mantener congelados los pesos de las dos VGG.

Para que los pesos de una VGG queden congelados, simplemente se recorren las capas de este modelo y, por cada una de ellas, se realiza lo siguiente:

$$layer.trainable = False \quad (1)$$

Donde *layer* representa cada capa de cada una de las dos VGG del modelo.

La última modificación, con respecto al bilineal con diferentes CNN, ha sido la sustitución del scheduler que se estaba utilizando hasta el momento, y sustituirlo por uno que modifique el *learning rate* en la *epoch* 150.

Los resultados obtenidos (con 500 *epochs*, Optimizador Adam y *learning rate* inicial 0.1) han sido:

Pesos de las VGG16	Resultados (%)
Sin congelar	38.78
Una sin congelar y otra congelados	57.82
Congelados	<b>63.01</b>

**Tabla 2:** Resultados del modelo bilineal con dos VGG16.

Como se puede apreciar, el mejor modelo aplicado en esta tarea de identificación de modelos de coches ha sido el que consta de dos VGG16 preentrenadas con los pesos congelados, alcanzando una precisión del **63.01** %.