



UNIVERSIDAD DE GRANADA  
MÁSTER DE CIENCIA DE DATOS E INGENIERÍA DE  
COMPUTADORES  
CURSO ACADÉMICO 2019-2020  
VISIÓN POR COMPUTADOR

## Clasificación y Segmentación de Peatones.

*Análisis, Aplicación y Comparación de modelos para  
clasificación y segmentación de imágenes con peatones.*

Nicolás Cubero

18 de Junio de 2020

# Índice

Índice de tablas	2
1. Introducción	3
2. Clasificación de peatones	3
2.1. Modelos basados en SVM con HOG . . . . .	4
2.1.1. Kernel lineal . . . . .	4
2.1.2. Kernel RBF . . . . .	5
2.1.3. Kernel polinómico . . . . .	7
2.2. Modelos CNN con <i>transfer learning</i> . . . . .	9
2.2.1. Experimentos <i>MobileNetV2</i> . . . . .	9
2.2.2. Experimentos <i>VGG16</i> . . . . .	11
2.2.3. Experimentos <i>ResNet50</i> . . . . .	12
2.3. Comparación del mejor modelo SVM con el mejor modelo CNN	13

## Índice de tablas

1.	Métricas de accuracy, precision, recall y área bajo la curva ROC del modelo SVM con kernel lineal sobre el conjunto de test. . . . .	4
2.	Métricas de accuracy, precision, recall y área bajo la curva ROC del modelo SVM con kernel rbf sobre el conjunto de test. . . . .	7
3.	Métricas de accuracy, precision, recall y área bajo la curva ROC del modelo SVM con kernel polinómico sobre el conjunto de test. . . . .	9
4.	Métricas de accuracy y loss medidas sobre los conjuntos de <i>train</i> y <i>test</i> ante diferentes valores de tamaño del lote . . . . .	10
5.	Métricas de métricas de <i>precision</i> , <i>recall</i> y <i>f1-score</i> medidas sobre el conjunto de <i>test</i> ante diferentes valores de tamaño del lote . . . . .	10
6.	Matriz de confusión del modelo basado en MobileNetV2 con batch size de 35 sobre el conjunto de test . . . . .	11
7.	Métricas de accuracy y loss medidas sobre los conjuntos de <i>train</i> y <i>test</i> ante diferentes valores de tamaño del lote obtenidos a partir de la red VGG16 . . . . .	11
8.	Métricas de métricas de <i>precision</i> , <i>recall</i> y <i>f1-score</i> medidas sobre el conjunto de <i>test</i> ante diferentes valores de tamaño del lote obtenidos mediante la red VGG16 . . . . .	12
9.	Métricas de accuracy y loss medidas sobre los conjuntos de <i>train</i> y <i>test</i> ante diferentes valores de tamaño del lote obtenidos a partir de la red ResNet50 . . . . .	13
10.	Métricas de métricas de <i>precision</i> , <i>recall</i> y <i>f1-score</i> medidas sobre el conjunto de <i>test</i> ante diferentes valores de tamaño del lote obtenidos mediante la red ResNet50 . . . . .	13
11.	Comparación entre el mejor modelo obtenido con clasificadores SVM y el mejor modelo obtenido con CNN . . . . .	14

# 1. Introducción

En este proyecto, se pretende llevar a cabo la aplicación de diversos modelos de *Machine Learning* aplicados al ámbito de la *Visión por Computador* para la resolución de uno de los problemas más clásicos en este área: La detección de peatones.

En concreto, se tratarán dos problemas relacionados con la detección de peatones: Por un lado, la clasificación de imágenes para determinar si aparece algún peatón en ella, y por otro, la segmentación de imágenes para la identificación de peatones presentes en la misma.

Para tratar el problema de clasificación, se hará uso de modelos basados en ***Support Vector Machine (SVM)*** haciendo uso de descriptores de ***Histogram of Oriented Gradient (HOG)*** y modelos basados en redes neuronales convolucionales que se construirán mediante *transfer-learning* partiendo de modelos conocidos.

Por último, se probarán diferentes configuraciones de redes neuronales de segmentación para determinar el modelo que ofrece una mayor bondad en la identificación de peatones en una imagen.

La ejecución de todos los experimentos se desarrollará mediante el lenguaje *Python* con la librería de visión artificial *OpenCV* y *Keras* con *Tensorflow* como *backend* para la construcción de los modelos de redes neuronales profundas.

# 2. Clasificación de peatones

Primeramente, se abordará el problema de la clasificación de peatones consistente en la identificación de imágenes en las que aparece algún peatón de aquellas en las que sólo aparece una imagen de fondo.

Se trata por lo tanto de un problema de clasificación binario que se resuelve, primeramente mediante un clasificador *SVM* haciendo uso de descriptores *HOG* para caracterizar y describir cada imagen y, también mediante redes neuronales convolucionales desarrolladas a partir de otras redes preexistentes (*AlexNet*, *VGG16*, *ResNet 18*, etc).

## 2.1. Modelos basados en SVM con HOG

La primera batería de experimentaciones se realiza con los clasificadores *SVM* haciendo uso de descriptores de HOG para caracterizar cada una de las imágenes. Para el cálculo de los descriptores de HOG se ha hecho uso de una implementación propia de este algoritmo realizada para este proyecto.

En el entrenamiento de los modelos se ha hecho uso de un *dataset* con imágenes de peatones e imágenes de fondo, de las cuales, se ha tomado una submuestra de 400 imágenes con peatones y se ha generado 400 imágenes de fondo, constituyendo un total de 800 imágenes.

De estas imágenes, el 80 % se ha empleado para el entrenamiento de los modelos reservando el 20 % restante para la evaluación (*test*) de los modelos. Por su parte, se han realizado diferentes experimentos probando diferentes *kernels* y parámetros de los modelos permitidos por cada *kernel*. Los resultados de este conjunto de experimentos se expondrán y analizarán a lo largo de toda la sección.

### 2.1.1. Kernel lineal

Con la función *kernel* lineal se probaron los siguientes valores de la constante regularizadora  $C$ :  $C \in \{0,001, 0,01, 0,05, 0,1, 1, 2, 5, 7, 10\}$

C	Test			
	accuracy	precision	recall	ROC
0.0001	0.475	0	0	0.5
0.001	0.475	0	0	0.5
0.005	0.475	0	0	0.5
0.01	0.475	0	0	0.5
0.05	0.99375	0.98824	1	0.99342
0.1	0.99375	0.98824	1	0.99342
0.5	0.99375	0.98824	1	0.99342
1	0.99375	0.98824	1	0.99342
2	0.99375	0.98824	1	0.99342
5	0.99375	0.98824	1	0.99342
7	0.99375	0.98824	1	0.99342
10	0.99375	0.98824	1	0.99342

Tabla 1: Métricas de accuracy, precision, recall y área bajo la curva ROC del modelo SVM con kernel lineal sobre el conjunto de test.

Los resultados de los experimentos reflejan rendimientos pésimos representados por clasificadores triviales para valores de la constante de regularización inferiores a 0.05.

Una vez se incrementa el valor de la constante a este valor y, también para valores superiores a este, se obtiene el máximo rendimiento de los clasificadores independientemente del incremento de este parámetro.

Se aprecia que este máximo rendimiento obtenido por los modelos con este kernel lineal resulta muy óptimo, acercándose a la totalidad de aciertos de ambas clases.

### 2.1.2. Kernel RBF

Se realiza una segunda experimentación considerando, en esta ocasión, la función *kernel* de base radial (*rbf*) y considerando el siguiente conjunto de valores para cada parámetro:

- $C \in \{0,001, 0,01, 0,05, 0,1, 0,5, 1, 2, 5\}$
- $\gamma \in \{10^{-3}, 10^{-2}, 0,1, 0,5, 1, 2, 5, 10\}$

Los resultados de esta experimentación se muestran a continuación:

C	gamma	CCR	precision	recall	ROC
$10^{-3}$	$10^{-3}$	0.41875	0.41875	1	0.5
	$10^{-2}$	0.41875	0.41875	1	0.5
	0.1	0.41875	0.41875	1	0.5
	0.5	0.41875	0.41875	1	0.5
	1	0.41875	0.41875	1	0.5
	2	0.41875	0.41875	1	0.5
	5	0.41875	0.41875	1	0.5
	10	0.41875	0.41875	1	0.5
$10^{-2}$	$10^{-3}$	0.41875	0.41875	1	0.5
	$10^{-2}$	0.41875	0.41875	1	0.5
	0.1	0.41875	0.41875	1	0.5
	0.5	0.41875	0.41875	1	0.5
	1	0.41875	0.41875	1	0.5
	2	0.41875	0.41875	1	0.5
	5	0.41875	0.41875	1	0.5
	10	0.41875	0.41875	1	0.5

0.05	$10^{-3}$	0.41875	0.41875	1	0.5
	$10^2$	0.41875	0.41875	1	0.5
	0.1	0.41875	0.41875	1	0.5
	0.5	0.7	0.5826	1	0.74193
	1	0.95625	0.9054	1	0.96237
	2	0.925	0.8481	1	0.93548
	5	0.41875	0.41875	1	0.5
	10	0.41875	0.41875	1	0.5
0.1	$10^{-3}$	0.41875	0.41875	1	0.5
	$10^2$	0.41875	0.41875	1	0.5
	0.1	0.41875	0.41875	1	0.5
	0.5	0.99375	0.98529	1	0.99462
	1	1	1	1	1
	2	0.99375	0.98529	1	0.99462
	5	0.41875	0.41875	1	0.5
	10	0.41875	0.41875	1	0.5
0.5	$10^{-3}$	0.41875	0.41875	1	0.5
	$10^2$	0.41875	0.41875	1	0.5
	0.1	0.99375	0.98529	1	0.99462
	0.5	1	1	1	1
	1	1	1	1	1
	2	1	1	1	1
	5	0.99375	0.98529	1	0.99462
	10	0.41875	0.41875	1	0.5
	$10^{-3}$	0.41875	0.41875	1	0.5
	$10^2$	0.41875	0.41875	1	0.5
1	0.1	1	1	1	1
	0.5	1	1	1	1
	1	1	1	1	1
	2	1	1	1	1
	5	0.99375	0.98529	1	0.99462
	10	0.96875	0.98438	0.94030	0.96477
	$10^{-3}$	0.41875	0.41875	1	0.5
	$10^2$	0.8875	0.78824	1	0.90323
	0.1	1	1	1	1
	0.5	1	1	1	1
2	1	1	1	1	1
	2	1	1	1	1

	5	0.99375	0.98529	1	0.99462
	10	0.96875	0.98438	0.94030	0.96477
	$10^{-3}$	0.41875	0.41875	1	0.5
	$10^2$	0.99375	0.98529	1	0.99462
	0.1	1	1	1	1
	0.5	1	1	1	1
	1	1	1	1	1
	2	1	1	1	1
5	5	0.99375	0.98529	1	0.99462
	10	0.96875	0.98438	0.94030	0.96477

Tabla 2: Métricas de accuracy, precision, recall y área bajo la curva ROC del modelo SVM con kernel rbf sobre el conjunto de test.

Los resultados de esta experimentación muestran un fenómeno similar al que se observó en los anteriores experimentos: sólo cuando la constante  $C$  se eleva a 0.05, los modelos ofrecen rendimientos aceptables, además se observa que el rendimiento óptimo se obtiene con valores de  $\gamma$  centrados en 1, de manera que al incrementar o decrementar el valor de este parámetro respecto de este valor, el rendimiento tiende a disminuir hasta dar lugar a clasificadores triviales

Por su parte, conforme se incrementa el valor del parámetro  $C$ , los rendimientos de los modelos con valores de  $\gamma$  diferentes a 1 también tienden a incrementarse a excepción de los valores de  $\gamma$  más alejados de 1. En cualquier caso, con el *kernel* rbf se obtienen modelos que ofrecen una bondad absoluta tanto en el conjunto de *train* como de *test*.

### 2.1.3. Kernel polinómico

Como último conjunto de experimentos, se usa una función *kernel* polinómico de forma que el conjunto de parámetros barridos es el siguiente:

- $C \in \{0,0001, 0,001, 0,01, 0,05, 0,1, 0,5, 1, 2, 5\}$
- $degree \in \{2, 3, 4\}$

Los resultados obtenidos se muestran en la siguiente tabla:



degree	C	CCR	precision	recall	ROC
2	$10^{-4}$	0.475	0	0	0.5
	0.005	0.475	0	0	0.5
	0.001	0.475	0	0	0.5
	0.05	1	1	1	1
	0.01	0.475	0	0	0.5
	0.5	1	1	1	1
	0.1	1	1	1	1
	1	1	1	1	1
	2	0.99375	0.98824	1	0.99342
	5	0.99375	0.98824	1	0.99342
	7	0.99375	0.98824	1	0.99342
	10	0.99375	0.98824	1	0.99342
3	$10^{-4}$	0.475	0	0	0.5
	0.005	0.475	0	0	0.5
	0.001	0.475	0	0	0.5
	0.05	1	1	1	1
	0.01	0.475	0	0	0.5
	0.5	1	1	1	1
	0.1	1	1	1	1
	1	1	1	1	1
	2	1	1	1	1
	5	1	1	1	1
	7	1	1	1	1
	10	1	1	1	1
4	$10^{-4}$	0.475	0	0	0.5
	0.005	0.475	0	0	0.5
	0.001	0.475	0	0	0.5
	0.05	0.9875	1	0.9762	0.9881
	0.01	0.475	0	0	0.5
	0.5	1	1	1	1
	0.1	1	1	1	1
	1	1	1	1	1
	2	1	1	1	1
	5	1	1	1	1
	7	1	1	1	1
	10	1	1	1	1

Tabla 3: Métricas de accuracy, precision, recall y área bajo la curva ROC del modelo SVM con kernel polinómico sobre el conjunto de test.

Con esta función de *kernel* polinómico, también se generaron algunos modelos que ofrecen la máxima bondad de clasificación tanto sobre el conjunto de *train* como sobre el conjunto de *test*.

## 2.2. Modelos CNN con *transfer learning*

A continuación, se tratará el desarrollo, evaluación y comparativa del rendimiento de modelos *CNN* entrenados haciendo uso de transferencia de conocimiento a partir de las redes *MobileNetV2*, *VGG16* y *Resnet50* sin el uso de ningún algoritmo de extracción de descriptores ni transformaciones previas a las imágenes.

Originalmente, para la realización de este conjunto de experimentos se propusieron las redes *AlexNet* y *ResNet18*, sin embargo, debido a que el *framework* ***keras*** no incluye ninguna de estas dos redes y tampoco fue posible encontrar una implementación no oficial acompañada de los pesos de la red original en la web y que no presentara errores y/o se haya quedado obsoleta, se decidieron usar en su lugar *MobileNetV2* y *ResNet50* que sí se encuentran incluidas en ***keras***.

### 2.2.1. Experimentos *MobileNetV2*

Se realiza una primera experimentación partiendo de la red preentrenada *MobileNetV2*, para la que se reemplazó la última capa de clasificación *softmax* por las siguientes capas:

1. Capa *fully-connected* de 1024 unidades con función de activación no lineal *RELU* (*Rectification Linear Unit*).
2. Dropout del 20 % entre las conexiones de esta capa y la siguiente.
3. Capa de clasificación con una unidad con función de activación no lineal sigmoide.

Como métrica de error se establece la función **entropía cruzada binaria** y también se evalúan como métricas adicionales, *accuracy*, *precision*, *recall*, *f1-score* así como la matriz de confusión.

Para todos los experimentos se establece un número de épocas de 25 y se barre el siguiente conjunto de valores de tamaño de lote (*batch size*):  $batch\_size \in \{10, 20, 35, 50, 70, 100\}$  obteniéndose los siguientes resultados:

batch size	Train		Test	
	accuracy	loss	accuracy	loss
10	0.51406	8.2040	0.50625	8.3735
20	0.71875	5.57323	0.71250	4.84491
35	0.82499	3.15977	0.85625	2.17534
50	0.50469	15.94021	0.5	15.5766
70	0.65156	6.15233	0.70625	4.96722
100	0.55316	9.94205	0.55	9.81510

Tabla 4: Métricas de accuracy y loss medidas sobre los conjuntos de *train* y *test* ante diferentes valores de tamaño del lote

Se muestran también las métricas de *precision*, *recall* y *f1-score* medidas sobre el conjunto de *test*:

batch size	precision	recall	f1-score
10	1	0.01220	0.02410
20	0.8	0.29268	0.42857
35	0.89655	0.63414	0.74285
50	1	0.02439	0.04761
70	0.92857	0.15853	0.27083
100	0.06098	0.55	0.11494

Tabla 5: Métricas de métricas de *precision*, *recall* y *f1-score* medidas sobre el conjunto de *test* ante diferentes valores de tamaño del lote

El mejor modelo se ha obtenido considerando un tamaño de lote de 35. Su matriz de confusión es la siguiente:

		Clase Real	
		+	-
Clase Predicha	+	5	0
	-	77	78

Tabla 6: Matriz de confusión del modelo basado en MobileNetV2 con batch size de 35 sobre el conjunto de test

### 2.2.2. Experimentos *VGG16*

Tomando la red *VGG16* se repite el anterior conjunto de experimentaciones realizadas en la anterior sección. Al igual que con la anterior, se reemplaza la última capa de clasificación *softmax* por las siguientes capas:

1. Capa *fully-connected* de 1024 unidades con función de activación no lineal *RELU* (*Rectification Linear Unit*).
2. Dropout del 20 % entre las conexiones de esta capa y la siguiente.
3. Capa de clasificación con una unidad con función de activación no lineal sigmoide.

Del mismo modo que en la anterior experimentación, se ejecuta el entrenamiento durante 25 épocas tomando el mismo conjunto de parámetros para el tamaño de lote *batch size*.

Los resultados obtenidos se muestran a continuación:

batch size	Train		Test	
	accuracy	loss	accuracy	loss
10	1	0	1	0
20	1	0	1	0
35	1	0	1	0
50	1	0	1	0
70	1	0	1	0
100	1	0	1	0

Tabla 7: Métricas de accuracy y loss medidas sobre los conjuntos de *train* y *test* ante diferentes valores de tamaño del lote obtenidos a partir de la red VGG16

batch size	precision	recall	f1-score
------------	-----------	--------	----------

10	1	1	1
20	1	1	1
35	1	1	1
50	1	1	1
70	1	1	1
100	1	1	1

Tabla 8: Métricas de métricas de *precision*, *recall* y *f1-score* medidas sobre el conjunto de *test* ante diferentes valores de tamaño del lote obtenidos mediante la red VGG16

En los anteriores resultados se observa que la bondad de los clasificadores entrenados es absoluta en las evaluaciones efectuadas.

### 2.2.3. Experimentos *ResNet50*

Por último, se ejecuta un último conjunto de experimentos tomando como partida la red *ResNet50* preentrenada.

Al igual que las anteriores se reemplaza la capa de clasificación por el siguiente conjunto de capas:

1. Capa *fully-connected* de 1024 unidades con función de activación no lineal *RELU* (*Rectification Linear Unit*).
2. Dropout del 20 % entre las conexiones de esta capa y la siguiente.
3. Capa de clasificación con una unidad con función de activación no lineal sigmoide.

Tomando el mismo conjunto de parámetros de entrenamiento que en los anteriores experimentos, se ejecutan los experimentos, obteniéndose los siguientes resultados:

batch size	Train		Test	
	accuracy	loss	accuracy	loss
10	0.99531	0.60553	0.97500	0.27402
20	0.99687	0.18859	1	0
35	0.99844	0.05486	0.98125	0.07440
50	0.99531	0.09262	1	0

70	1	0.00057	1	0.00023
100	0.99844	0.01424	0.99375	0.17079

Tabla 9: Métricas de accuracy y loss medidas sobre los conjuntos de *train* y *test* ante diferentes valores de tamaño del lote obtenidos a partir de la red ResNet50

batch size	precision	recall	f1-score
10	0.96429	1	0.98182
20	1	1	1
35	1	1	1
50	1	0.98765	0.99379
70	1	1	1
100	0.98780	1	0.99387

Tabla 10: Métricas de métricas de *precision*, *recall* y *f1-score* medidas sobre el conjunto de *test* ante diferentes valores de tamaño del lote obtenidos mediante la red ResNet50

A grandes rasgos, los modelos desarrollados a partir de *ResNet50* proveen una tasa de aciertos absoluta en la mayor parte de los experimentos. De los anteriores modelos se decide proponer el modelo obtenido con *batch size* 70 como el mejor por su absoluto *accuracy* obtenido tanto sobre el conjunto de *train* y *test*.

Los experimentos realizados muestran una bondad casi absoluta en todos los modelos desarrollados: Todos los modelos entrenados a partir de *VGG16* muestran una bondad absoluta seguido de los modelos basados en *ResNet50*, donde la mayoría de estos modelos también ofrecen una bondad absoluta sobre los conjuntos de *train* y *test*. Los modelos basados en *MobileNetV2* ofrecen por consiguiente, los rendimientos más bajos obtenidos aunque en sí constituyen valores muy óptimos.

### 2.3. Comparación del mejor modelo SVM con el mejor modelo CNN

Para acabar, se discute brevemente una comparación entre los rendimientos de los mejores modelos obtenido con *SVM* haciendo uso de descriptores *HOG* con los mejores modelos obtenidos con redes neuronales convolucionales.

les, que en este caso, se han obtenido partiendo de la red *VGG16*.

En la siguiente tabla se muestra de forma más detallada esta comparación:

modelo	accuracy	precision	recall
SVM	1	1	1
CNN	1	1	1

Tabla 11: Comparación entre el mejor modelo obtenido con clasificadores SVM y el mejor modelo obtenido con CNN

Ambos modelos reflejan los mejores resultados posibles por lo que, a efectos de bondad de clasificador resultaría indiferente escoger uno u otro. Sin embargo, a efectos prácticos y de la utilización directa de estos clasificadores para una aplicación real, resltaría más óptimo, desde el punto de vista del coste computacional, el modelo basado en *SVM*, que tomaría menos recursos de espacio y memoria para realizar la tarea de clasificación con el mismo rendimiento que una red CNN.