

Analyzing the effect of hyperparameters in a automobile classifier based on convolutional neural networks

Elian Laura Riveros

Universidad Nacional de San Agustín
Arequipa, Perú

Email: elian.laura.riv@gmail.com

José Galdos Chávez

Universidad Católica San Pablo
Arequipa, Perú

Email: jose.galdos.chavez@ucsp.edu.pe

Juan C. Gutiérrez Cáceres

Universidad Nacional de San Agustín
Universidad Católica San Pablo
Arequipa, Perú

Email: jcgutierrezc@gmail.com

Abstract—In the recent years the convolutional neural network is used successfully in applications of image classification, due to its deep and hierarchical architecture. The hyperparameters of the convolutional neural networks are of great influence to obtain good results in binary classification without the need of a large number of layers. The activation function, the weights initialization and the subsampling function are the three main hyperparameters. In the present work 27 models of convolutional neural network are trained and tested with automobile images taken from a surveillance camera. The illumination intensity of the test images are different from the training images, because they were taken from scenes of day, evening and night. We also demonstrate the influence of the mean of the images and the size of the filter kernel. The convolutional neural network model with the best result reached 95.6% of accuracy. The results of experiments show that neural networks predict successfully automobile images with varied illumination intensities overcome the techniques Haar Cascade and the Support Vector Machine.

Keywords—Image processing; convolutional networks; image classification; automobile recognition in images.

I. INTRODUCCIÓN

En los últimos años el enfoque de aprendizaje profundo a conseguido mayor auge, su funcionamiento en varios niveles de abstracción y representación ayuda a dar sentido a la información de la imagen [1], como parte de este enfoque surgieron las redes neuronales convolucionales (CNN), que tienen la capacidad de clasificar todo tipo de objetos con resultados prometedores y mejores aún cuando la distribución y el espacio de características del conjunto de datos de entrenamiento es la misma del conjunto de datos de prueba.

Las capas convolucionales y capas de convergencia de una CNN [2] constituyen la parte profunda y jerárquica de su arquitectura, convirtiéndola en un extractor de características invariante a la rotación, traslación y escala. Las últimas capas de una CNN constituyen lo que es una red neuronal completamente conectada sin capas ocultas, de esta manera una CNN se convierte en una herramienta de extracción y clasificación de características en un solo paso.

Las CNN a gran escala han logrado excelentes tasas de reconocimiento con más de 1 millón de imágenes agrupadas en 1000 clases [3], también se demostró su eficiencia en otros conjuntos de datos capturados con cámaras montadas

en vehículos en movimiento, son pocos los conjuntos de datos de imágenes extraídos de cámaras de videovigilancia, estos dispositivos han alcanzado un uso de gran impacto para la seguridad pública, por ser adecuables a los centros públicos y capaces de soportar grabaciones por varias horas. Es por ello que surgen técnicas para la clasificación de objetos a través de imágenes extraídas de una cámara de videovigilancia. La máquina de vectores de soporte(SVM) ha sido empleada por años como clasificador debido a su robustez ante muchas aplicaciones, en conjunto con un extractor de características como el histograma orientado a gradiente (HOG). La cascada de Haar llamada así por su aprendizaje en cascada de características tipo Haar, también ha sido parte de los grandes éxitos en clasificación, su aplicación como detector se fortalece gracias a la técnica de ventana piramidal deslizante la cual recorre toda la imagen en diferentes escalas para la búsqueda del objeto. Posteriormente la aparición del aprendizaje profundo se destacó por su aplicación exitosa en la clasificación de objetos. La seguridad ciudadana es uno de los tópicos de entorno público más discutidos, la búsqueda de soluciones para alcanzar la máxima seguridad es el reto de cada día, así también para el campo de investigación científica aplicada al procesamiento de imágenes. Algunas técnicas de clasificación de peatones, vehículos, rostros, etc, han acudido a la utilización de las CNN para sistemas aplicados al entorno público a través de imágenes de cámaras de videovigilancia.

Uno de los problemas que afronta la clasificación de imágenes de cámaras de videovigilancia es que el objeto sufre transformaciones durante su aparición en el video, tales como multiescala, diferencias de iluminación y variedad de perspectivas, resultando un conjunto de datos con diversidad de contextos. Motivados por este problema es que proponemos un modelo de CNN, para clasificación binaria, basado en un modelo con menor número de capas que los modelos recientes de gran escala, GoogleNet(22 capas) [4], VGGNet(16 capas) [5] y Alexnet(8 capas) [3]. Nuestro modelo propuesto es obtenido como resultado del análisis de la combinación de tres hiperparámetros y de alterar ligeramente los tamaños del kernel de filtro y el número de capas. Los experimentos también incluyen datos con diferentes niveles de iluminación del día, así también en modo infrarrojo, con la finalidad de demostrar la robustez y ventajas de las CNN ante diversos contextos de iluminación.

II. TRABAJOS RELACIONADOS

Los trabajos relacionados al reconocimiento de automóviles en visión se basan en las características que representan la apariencia del objeto tales como simetría, borde y sombras proyectadas [1], estas características posteriormente entran a un clasificador previamente entrenado para n clases de automóviles o solamente dos clases, automóviles y fondo. Durante muchos años un reconocimiento robusto de automóviles es comúnmente resuelto en dos etapas: la extracción de características y el aprendizaje supervisado. Las características obtenidas con la gradiente orientada a objetos(HOG), los patrones binarios locales(LBP) y las características de Haar(HAAR) son extremadamente acudidas en la literatura. Además de la transformada de características invariante a escala (SIFT), las características robustas de alto rendimiento (SURF) y la combinación de estas características son también aplicadas en la representación de la imagen de un automóvil. La máquina de vectores de soporte (SVM), Adaboost en cascada y redes neuronales, son algunas de las técnicas de reconocimiento que han demostrado buenos resultados en las pruebas, combinadas con las técnicas mencionadas de extracción de características.

El aprendizaje profundo a conseguido mayor auge en los últimos años, debido a sus procesamiento detalle a detalle de manera jerárquica, alcanzando así una alta abstracción de las características de la imagen. Wang *et al.* [1] proponen un detector de vehículos basado en deep belief network (DBN) con 2 capas escondidas, siendo la arquitectura que resultó con mejores resultados a comparación del uso de una capa escondida y otra de 3 capas escondidas. Utiliza un conjunto de imágenes de la parte posterior de vehículos, extraídos de la base de datos Caltech1999 [6] este es complementado con imágenes propias del autor. Las CNN también forman parte del aprendizaje profundo, se caracterizan por su capacidad de reconocer objetos abarcando las dos etapas, extracción y aprendizaje en un solo paso. El trabajo de Li *et al.* [7] propone adaptar un detector de vehículos a un dominio distinto, entrena una CNN con un conjunto de datos determinado, el modelo obtenido es capaz de detectar el mismo objeto en cualquier otro dominio. Dos detectores fueron usados, uno para detectar vehículos de perfil y otro para vehículos desde una perspectiva lateral.

El enfoque CNN también fue aplicado por Cai *et al.* [8] para la detección de automóviles donde resalta el uso de visión monocular y de visión binocular con CNN, nuevamente su conjunto de datos consiste en imágenes de la parte posterior de vehículos, muy parecidos a Caltech1999, aunque no lo menciona, su mejora en el tiempo de procesamiento y la tasa de detección se demuestra comparando su técnica con las de otros autores.

El algoritmo de AdaBoost también es un clasificador fuertemente acudido, una CNN también es utilizada para extraer el vector de características que posteriormente ingresan a un AdaBoost para obtener el modelo de detección de vehículos, como es el caso del trabajo de Song *et al.* [9], sus imágenes de entrenamiento son de dimensión 96x96px, a través de la experimentación el autor demuestra que 2 capas de convolución y 2 de submuestreo lanza un menor porcentaje de error que una arquitectura de 1 sola capa de convolución y 1 de submuestreo, sus resultados son comparados con SVM+HOG, Gabor+SVM y con el uso completo de una CNN.

En la investigación de He *et al.* [2] también se extrae las características de 8 clases de vehículos con una CNN, y un SVM es usado como clasificador. El conjunto de imágenes para el entrenamiento son extraídos de ImageNet [10]. Todos los trabajos de investigación en reconocimiento de automóviles por imágenes con aprendizaje profundo hacen uso de un conjunto de imágenes con el objeto automóvil en perspectiva frontal/posterior o lateral haciendo uso de una cámara montada en otro vehículo de la escena. La iluminación de la escena es constante en sus casos de prueba. Algunos autores experimentan en escenas públicas pero sin resultados concretos.

III. RED NEURONAL CONVOLUCIONAL

El aprendizaje profundo es una clase más del mundo del aprendizaje de máquina, con una arquitectura jerárquica. Se realiza en múltiples niveles de abstracción, transformación y representación, partiendo de características de bajo nivel hasta alcanzar el aprendizaje de características de alto nivel. Este enfoque permite capturar y dar sentido a la información de la imagen. Las CNN forman parte del aprendizaje profundo, están inspiradas en las redes neuronales y estas a su vez están inspiradas en la interacción compleja entre neuronas biológicas, donde participan las dendritas, los axones y su sinapsis.

La profundidad en las CNN es muy importante para un buen reconocimiento, suprime la atención en el aumento de capas concentrándose en hacerla profunda por el número de neuronas, es decir, procesando detalle a detalle de manera jerárquica, aprendiendo características cada vez más complejas. En la Figura 1 se puede visualizar la descomposición de características de la imagen de un automóvil, durante el aprendizaje de esta imagen en la primera capa se debe aprender a detectar bordes, en la segunda se aprende a descomponer los bordes en esquinas, la siguiente capa aprende texturas, de esta manera la red puede hacer predicciones más precisas.

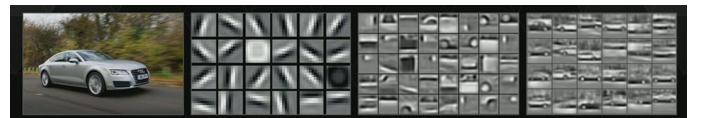


Figura 1: Descomposición jerárquica de una imagen con una arquitectura de aprendizaje profundo.

III-A. Arquitectura de una CNN

Existen variedad de arquitecturas para construir una CNN, cada una aplicada a diversos casos de estudio. AlexNet es la arquitectura que consta de 60 millones de parámetros, para clasificar 1.3 millones de imágenes pertenecientes a 1000 clases, GoogleNet consta de 22 capas y reduce la cantidad de parámetros a 40 millones y también fue diseñada para una clasificación a gran escala. Una de las primeras aplicaciones existosas de redes convolucionales fue desarrollada por Yann LeCun en 1998, se trata de LeNet-5, y fue utilizada para reconocer dígitos, códigos comprimidos, entre otros.

La CNN se constituye de 2 partes: 1) Extractor automático de características multiestado. Cada estado consta de una capa convolucional y otra de submuestreo, 2) Un clasificador, que es una red neuronal completamente conectada sin capas ocultas.

A pesar de las numerosas arquitecturas de una CNN los componentes básicos permanecen en todas. Una CNN típicamente consiste de tres tipos de capas, la capa convolucional, la capa de submuestreo y la capa completamente conectada. La Figura 3 muestra la arquitectura de LeNet-5. El objetivo de una capa convolucional es aprender la representación de características. Cada neurona se encarga de una submuestra de imagen(mapa de características), que al inicio será la imagen de entrada completa. La convolución se realiza entre cada mapa de características y un kernel que contiene valores aprendidos(pesos). También se hace uso de la propiedad de pesos compartidos(o replicación de pesos) que tiene la ventaja de reducir la complejidad del modelo y facilitar la etapa de entrenamiento. Las características no-lineales son rescatadas con la función de activación, que introduce la no-linealidad en la CNN.

En la capa de submuestreo se consigue la invarianza espacial para reducir la resolución de los mapas de características, de esta manera la CNN se hace más ligera en la etapa de entrenamiento, rescatando las características más relevantes a través de alguna función de submuestreo. Cada mapa de características de la capa de submuestreo es conectada a su correspondiente mapa de características de la capa convolucional precedente.

Luego de las capas de convolución y submuestreo se presenta una o más capas completamente conectadas, igual a un perceptrón multicapa. La clasificación de una CNN se lleva a cabo con un aprendizaje supervisado que puede ser visto en tres pasos principales: *feedforward*, *backward* y *update*, dirigidos por el método de la Gradiente Descendiente Estocástica (Stochastic Gradient descent, SGD). El método SGD calcula la derivada en cada dato de entrenamiento y realiza la actualización inmediatamente. Antes que el paso *backward* sea inicializado se reporta la función de error, en el presente trabajo hacemos uso de la función de regresión *Softmax* que genera una distribución de probabilidad de los valores de salida. En la clasificación también es común el uso de otro método, como SVM usado conjuntamente con CNN, esta propuesta fue elegida por [2].

III-B. Hiperparámetros de una CNN

El diseño de arquitecturas basadas en redes neuronales funcionan bajo un conjunto de hiperparámetros cuyas combinaciones influyen drásticamente en la tasa de precisión de sus resultados. La cantidad de datos de entrenamiento, el número de clases de los objetos a entrenar, el tamaño de las imágenes de entrada, son algunos de los aspectos que influyen a la hora de diseñar una arquitectura de aprendizaje profundo orientado a visión. En la tabla I se describen tres hiperparámetros que serán utilizados para las pruebas en la búsqueda del mejor modelo de CNN para detección de automóviles.

IV. IMPLEMENTACIÓN

La librería Caffe[11] brinda la posibilidad de usar un framework que engloba la implementación de modelos de aprendizaje profundo. Su código fuente es publicado en www.github.com/BVLC/caffe. Entre los modelos de CNN que ofrece Caffe están LeNet-5, Imagenet, GoogleNet, AlexNet, entre otros. Tiene la configuración para ser ejecutado sobre

Tabla I: Hiperparámetros de una CNN

Inicialización de pesos	
Distribución Uniforme	$x \sim U(a,b)$ intervalo [a,b]
Distribución de Gauss	$x \sim N(\theta, \delta)$ media θ , desviación estándar δ
Algoritmo Xavier	$r = \sqrt{\frac{6}{n_{in} + n_{out}}}$ conexiones entrantes a la neurona, (n_{in}) conexiones salientes de la neurona (n_{out})
Función de Activación	
Función Sigmoid	$\sigma(x) = \frac{1}{(1+e^{-x})}$ restringida entre 0-1
Función ReLU	$relu(x) = max(x, 0)$ Discontinuidad en cero
Función PRELU	$prelu(x_i) = max(0, x_i) + a_i min(0, x_i)$ a_i es aprendido por el canal i
Función de submuestreo	
Submuestreo máximo (MAX)	$s_j = \max_{i \in R_j} a_i$ sea R una región
Submuestreo promedio (AVE)	$s_j = \frac{1}{ R_j } \sum_{i \in R_j} a_i$
Submuestreo estocástico (STO)	$p_i = \frac{a_i}{\sum_{k \in R_j} a_k}$ probabilidades p de cada región j

CPU o sobre GPU. La ejecución de la librería caffe para el presente experimento se ha realizado en una computadora de 8 núcleos, con 8GB de RAM, 2.7 Ghz, conteniendo una tarjeta gráfica nVidia GeForce GT 750M. Los experimentos con la técnica cascada de haar fueron implementados con la librería OpenCV [12](Intel Open Source Computer Vision Library), entre sus módulos se adquieren funciones para medir el rendimiento de clasificadores tipo cascada, y otras funciones para el procesamiento de imágenes.

V. EXPERIMENTOS Y RESULTADOS

V-A. Datos de entrenamiento y prueba

El conjunto de muestras positivas y negativas, tanto para la etapa de entrenamiento como para la etapa de pruebas, fue extraído de los videos obtenidos con una cámara de videovigilancia [13], la cual se encuentra ubicada en la vía pública, aproximadamente a 4mts. de altura sobre la acera, en los exteriores de una universidad.

Algunas imágenes de muestras para el entrenamiento se observan en la tabla II. El entrenamiento fue realizado con 16800 imágenes, siendo 8400 muestras positivas y 8400 muestras negativas. Estas imágenes fueron extraídas de grabaciones hechas a medio día, bajo una alta intensidad de la iluminación del sol.

Los datos de prueba fueron divididos en tres conjuntos. El primer conjunto de datos de prueba, al que denominaremos CD11, consta de 6000 imágenes obtenidas entre las 7:00am y 5:00pm, siendo 3000 imágenes positivas y 3000 imágenes negativas. Las imágenes obtenidas en este rango de horas presentan variadas intensidades de iluminación, que se puede apreciar en la Figura 2a y en la Figura 2b, a diferencia de

Tabla II: Imágenes para el entrenamiento del clasificador CNN.

Muestras positivas	Muestras negativas

las imágenes de entrenamiento que presentan un solo nivel de iluminación.

El segundo conjunto de datos de prueba, denominado CD21, es captado en horas del atardecer y el tercer conjunto de datos al que llamaremos CD22, fue recolectado de una grabación nocturna con el modo infrarrojo de la cámara de videovigilancia. Una muestra de las escenas de los conjuntos CD21 y CD22 se pueden observar en la Figura 2c y en la Figura 2d respectivamente. Cada uno, CD21 y CD22, consta de 2200 imágenes conformado por 1100 imágenes positivas y 1100 imágenes negativas.

Cada imagen de los datos de prueba son etiquetadas con el valor 0 si es imagen de fondo y con el valor 1 si es imagen de automóvil, estas etiquetas son denominadas etiquetas reales. Las etiquetas obtenidas por el clasificador las denominamos etiquetas de predicción. Las etiquetas de predicción son comparadas con las etiquetas reales a través de la fórmula de la precisión (1), donde VP: Verdaderos positivos, VN: Verdaderos negativos, P: Positivas, N: Negativos. VP y VN pertenecen a las etiquetas de predicción, P y N pertenecen a las etiquetas reales, esto quiere decir que $P + N = \text{conjunto total de datos de prueba}$.

$$\text{Precisión} = \frac{VP + VN}{P + N} \quad (1)$$

V-B. Arquitecturas de CNN para el reconocimiento de automóviles

Las arquitecturas de aprendizaje profundo tienen una fuerte dependencia hacia los datos entrenados, y las características aprendidas no pueden transferirse a predecir datos diferentes a los datos entrenados [14]. Una CNN se conforma principalmente de kerneles que se encargan de procesar cada imagen para la extracción de características, alguna modificación de parámetros en el diseño de una CNN puede influir en el reconocimiento de un conjunto de datos de un determinado contexto. Es por ello que a continuación se realizan tres experimentos para alcanzar el mejor resultado de reconocimiento de automóviles obtenidos de una cámara de videovigilancia.

1) *Primer experimento:* LeNet-5 es una de las arquitecturas de CNN más sencilla y con resultados exitosos, con solo 5 capas ha logrado un alto porcentaje de reconocimiento de dígitos y letras manuscritas de la base de datos MNIST [15].

Algunos clasificadores de objetos se basaron en esta arquitectura, haciendo ligeras modificaciones en los hiperparámetros y en el número de capas [7] [16].

El primer experimento consta de tres pruebas, realizadas con nuestro conjunto datos de entrenamiento y el conjunto de datos de prueba CD11, ambos conjuntos descritos en la sección V-A.

En la primera prueba se realizó el entrenamiento del modelo original de CNN LeNet-5, con la finalidad de observar la eficiencia de este modelo para el reconocimiento de automóviles. El modelo original de LeNet-5 consta de la función de activación ReLU, la técnica de inicialización de pesos Xavier y de la función de submuestreo MAX.

En la segunda prueba resaltamos la importancia de calcular el valor de la media como un parámetro adicional. El valor de la media permite la normalización del brillo en el conjunto de datos de entrenamiento, si dos imágenes tienen el mismo contenido pero con una ligera diferencia de brillo entonces, con el parámetro de la media estas dos imágenes se verían iguales. Se obtienen mejores resultados de precisión cuando el parámetro de la media es incluido en el entrenamiento del modelo LeNet-5.

La tercera prueba realizada en LeNet-5 nos demuestra que los resultados de predicción varían según el tamaño de *batch*, el valor de este parámetro es la cantidad de imágenes que pasan por la red en cada iteración, un tamaño de batch igual a la cantidad de imágenes significaría que los pesos de la red se actualizarían una sola vez, lo que no permitiría una rápida convergencia a la función objetivo, un tamaño de batch menor (32, 64, 100, etc) significaría más actualizaciones de los pesos, es decir mayor cantidad de iteraciones, hasta que se alcanza el aprendizaje de los datos de entrenamiento, pero a su vez implicaría una mayor varianza del error, por otro lado un posible *overfitting*. La influencia del tamaño del *batch* en los resultados es dependiente de la cantidad de datos, decir que un tamaño de *batch* más grande es mejor no aplica a todos los casos, lo recomendable es realizar pruebas con un rango de valores de batch para el porcentaje de predicción. No existe un valor correcto para el *batch*, por ello las arquitecturas estándar de CNN no coinciden en ese valor, los autores han experimentado para encontrar el valor convencional a su clasificación. No hay un estudios dedicados a la influencia del parámetro *batch* en las CNN, es así que consideramos por conveniente hacer pruebas para encontrar un valor que incremente los resultados de predicción. Los resultados del primer experimento se pueden ver en la tabla III, donde se ha experimentado con el parámetro de la media y con un rango de valores de *batch* que fueron utilizados en las arquitecturas estándar: 32 (GoogleNet), 64 (Lenet), 100(CIFAR) y 256(AlexNet).

Según la tabla III los porcentajes de predicción con los valores 100 y 256 son superiores, la diferencia del resultado entre estos dos valores es mínima, hemos decidido usar el valor 100 para los siguientes experimentos debido a que el uso del valor 256 aumenta ligeramente el tiempo de entrenamiento.

2) *Segundo experimento:* En el segundo experimento diseñamos 27 arquitecturas de CNN, obtenidas de la combinación de los tres hiperparámetros descritos en la sección III-B: función de activación, inicialización de pesos y función de submuestreo. Se ha considerado 100 como tamaño de

Tabla III: Resultados del experimento 1, realizado con el modelo LeNet-5 para el reconocimiento de imágenes con automóviles.

Prueba	Mean	Batch	Precisión
1	no	64	92.17 %
2	yes	64	93.11 %
3	yes	32	92.53 %
4	yes	100	94.17 %
5	yes	256	94.26 %

Tabla IV: Resultados obtenidos con el segundo experimento.

CNN	FA	IP	FS	K3	K5	K3-3	K5-3
M1	Sigmoide	Xavier	Ave	92.22 %	92.97 %	92.57 %	93.03 %
M2	Sigmoide	Xavier	Max	94.27 %	95.52 %	94.32 %	94.88 %
M3	Sigmoide	Xavier	Sto	51.12 %	50.30 %	-	-
M4	Sigmoide	Uniforme	Ave	92.55 %	89.87 %	92.62 %	92.62 %
M5	Sigmoide	Uniforme	Max	91.87 %	90.63 %	94.37 %	91.95 %
M6	Sigmoide	Uniforme	Sto	49.97 %	50.00 %	-	-
M7	Sigmoide	Gaussian	Ave	82.70 %	83.20 %	50.00 %	50.00 %
M8	Sigmoide	Gaussian	Max	85.32 %	90.62 %	50.00 %	50.00 %
M9	Sigmoide	Gaussian	Sto	68.20 %	72.68 %	-	-
M10	ReLU	Xavier	Ave	94.23 %	93.38 %	93.92 %	93.92 %
M11	ReLU	Xavier	Max	95.60 %	93.75 %	-	-
M12	ReLU	Xavier	Sto	50.53 %	50.05 %	-	-
M13	ReLU	Uniforme	Ave	91.82 %	94.25 %	94.38 %	94.38 %
M14	ReLU	Uniforme	Max	93.65 %	89.25 %	93.35 %	93.35 %
M15	ReLU	Uniforme	Sto	50.22 %	49.85 %	-	-
M16	ReLU	Gaussian	Ave	93.28 %	92.93 %	50.00 %	50.00 %
M17	ReLU	Gaussian	Max	93.37 %	92.78 %	50.00 %	50.00 %
M18	ReLU	Gaussian	Sto	53.83 %	56.80 %	-	-
M19	ReLU	Xavier	Ave	94.50 %	93.67 %	95.37 %	95.37 %
M20	PReLU	Xavier	Max	95.13 %	94.13 %	95.12 %	95.12 %
M21	PReLU	Xavier	Sto	49.87 %	50.70 %	-	-
M22	PReLU	Uniforme	Ave	91.75 %	90.05 %	93.53 %	93.53 %
M23	PReLU	Uniforme	Max	92.58 %	88.55 %	91.12 %	91.12 %
M24	PReLU	Uniforme	Sto	50.55 %	50.13 %	-	-
M25	PReLU	Gaussian	Ave	93.05 %	91.52 %	50.00 %	50.00 %
M26	PReLU	Gaussian	Max	92.90 %	94.58 %	50.00 %	50.00 %
M27	PReLU	Gaussian	Sto	55.18 %	57.50 %	-	-

batch y se ha incluído el parámetro de la media en todas las arquitecturas. Se han realizado en total 4 pruebas que se diferencian en el tamaño del kernel de convolución y en la cantidad de capas convolucionales y de submuestreo, con la finalidad de observar la influencia de estos cambios en una CNN. La tabla IV contiene las siguientes columnas: modelo de CNN (CNN), función de activación(FA), técnica de inicialización de pesos(IP), función de submuestreo(FS), precisión obtenida con kernel 3x3(K3), precisión obtenida con kernel 5x5 (K5), arquitectura con 3 capas convolucionales de kernel 3x3 y 3 capas de submuestreo (K3-3), y en la última columna los resultados obtenidos con una arquitectura de 3 capas convolucionales de kernel 5x5 y 3 capas de submuestreo(K5-3). Los resultados obtenidos en K3 y K5 fueron realizados bajo la arquitectura original de LeNet-5, es decir dos capas convolucionales y 2 capas de submuestreo.

Después de analizar los resultados de la tabla IV observamos que los porcentajes se asemejan, siendo mayor a 95 %. Entre usar un kernel de 3x3 y un kernel de 5x5 la diferencia es mínima, 0.08 %, esto se puede observar en los modelos M11 y M2 respectivamente. Ambos modelos utilizan Xavier y Max, únicamente se diferencia en la función de activación, M11 fue entrenado con ReLU y M2 con Sigmoide. También podemos observar que la técnica Xavier para la inicialización de pesos se encuentra en los modelos de mejor porcentaje de precisión, resaltados con un color negro más intenso.

En las dos últimas columnas de la tabla IV no se hicieron las pruebas con aquellos modelos de función de submuestreo estocástico porque en las columnas K3 y K5 se observa que se

Tabla V: Comparación en la precisión obtenida con CNN y otras técnicas.

Escenas	CNN-M11	C-Haar	SVM
CD11	95.60 %	88.05 %	81.25 %
CD21	88.00 %	81.95 %	59.68 %
CD22	66.00 %	74.59 %	64.54 %

obtienen los peores porcentajes, menor a 80 % con esta técnica, Sto. Entre K3-3 y K5-3 no presenta diferencias resaltantes, en casi todos los casos de modelos los porcentajes de precisión son los mismos, siendo el mejor el del modelo M19 con 95.37 %. Esto nos permite deducir el tamaño del kernel no es de gran influencia en las predicciones de un modelo CNN basado en LeNet-5.

3) *Tercer experimento:* Despues de haber realizado dos experimentos, concluimos que el mejor modelo de CNN para nuestro reconocedor de automóviles es aquel que se entrena con el parámetro de la media, con un tamaño de batch igual a 100, un kernel de 3x3 para cada convolución, y una arquitectura de capas igual a LeNet-5, es decir, 2 capas convolucionales y 2 capas de submuestreo. Los resultados de este conjunto de características se observan en la columna K3 de la tabla IV, donde el mejor porcentaje es el obtenido por M11, cuyos hiperparámetros coinciden con los hiperparámetros de LeNet-5 original, que son ReLU, Xavier y Max.

En la Figura 3 esquematizamos la arquitectura del mejor modelo CNN obtenido en los experimentos anteriores. La capa convolucional(convolutional layer) es nombrada CLx, la capa de submuestreo(subsampling layer) es nombrada SLx, y una capa completamente conectada(fully connected layer) se abreviará como FCLx. La imagen de entrada es de 44px de ancho y 28px de alto debido a que los automóviles de las imágenes tienen una postura diagonal, como se puede ver en las muestras de la tabla II. En la capa CL1 se realiza la convolución resultando imágenes de 40x24px, su reducción de tamaño se debe al kernel del filtro que se observa en la Figura 4. Seguidamente la capa de submuestreo SL2 reduce al 50 % la imagen con una función de submuestreo, la capa CL3 obtiene por segunda vez las imágenes convolucionadas que serán reducidas en la capa SL4, la capa CL5 realiza una convolución y a la vez actúa como una capa completamente conectada, a través de una función de activación. Se conecta a la siguiente capa completamente conectada FCL6. Finalmente se obtiene un valor que indica la etiqueta predicha para la imagen de entrada, si el valor de predicción es '1' se trata de la imagen de un automóvil, si el valor es '0' entonces la imagen no es un automóvil.

Como parte del tercer experimento, utilizamos el modelo M11 para los conjuntos de datos CD21 y CD22, descritos en la sección V-A, que contienen una menor intensidad de iluminación en comparación al conjunto CD11. Los resultados se observan en la tabla V. La columna CNN-M11 contiene los resultados de precisión obtenidos con el modelo M11, C-Haar representa a una Cascada con Haar, y SVM representa a una máquina de vectores de soporte. Para el presente experimento los clasificadores C-Haar y SVM fueron entrenados con el mismo conjunto de muestras con que fue entrenado CNN-M11.

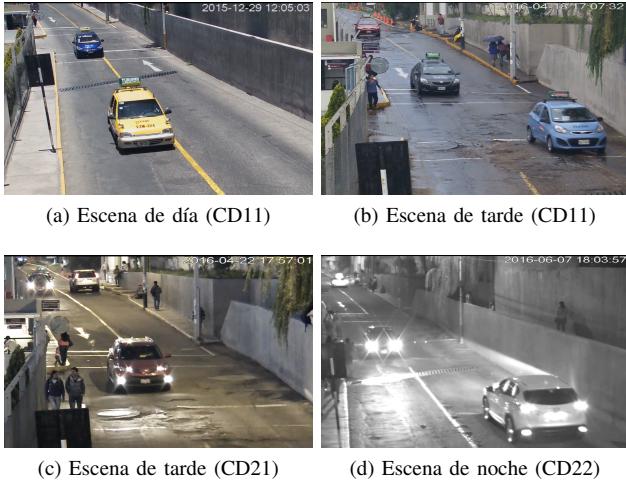


Figura 2: Escenas que demuestran la variación de iluminación para el conjunto de datos de prueba.

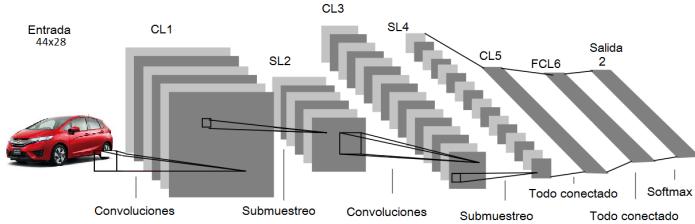


Figura 3: Arquitectura LeNet-5 para el reconocimiento de automóviles. Representación basada en [15].

V-C. Evaluación de hiperparámetros

La mayor precisión obtenida en el modelo M11 nos demuestra que el uso de ReLU es de rendimiento satisfactorio tal y como lo experimentaron los autores de [3] para una clasificación a gran escala. En cuanto a la función de activación sigmoide fue analizada por [18] en comparación con la función tangente hiperbólica. El autor concluye que la función sigmoide debe ser evitada cuando se inicializa los pesos con valores muy pequeños, ya que la curva de aprendizaje no es favorable en el rendimiento del clasificador, según nuestros experimentos, la función sigmoide junto a la técnica Xavier y la función MAX conforman el mejor modelo de kernel 5x5, tabla IV.

El algoritmo de Xavier hace posible la distribución uniforme escalada, el autor del algoritmo [18] demostró su eficiencia con una parte del conjunto de datos ImageNet, también con imágenes de formas, entre otras.

La técnica de submuestreo MAX logra una rápida tasa de convergencia seleccionando las características invariantes que mejoran el rendimiento de generalización [19]. El submuestreo con MAX activa las regiones de tamaño (k_x, k_y), que se deslizan sin sobreposición por la imagen, reduciéndola por un factor de k_x y k_y en cada dirección.

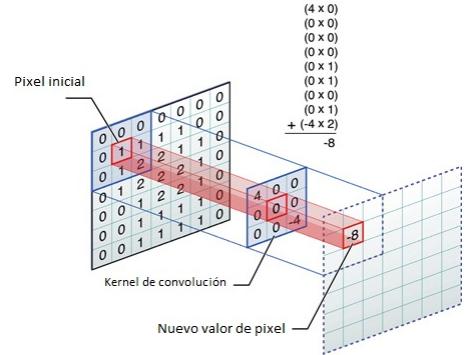


Figura 4: Convolución con una plantilla(o kernel) de dimensión 3x3. Imagen basada en [17].

V-D. Comparación de CNN con otros clasificadores

Por los experimentos en CNN reportados en la literatura sabemos que las arquitecturas GoogleNet(22 capas) [4], Alexnet(8 capas) [3] y entre otras, han logrado grandes resultados en detección de objetos con más de 10 categorías, su robustez es resultado de su gran profundidad es decir mayor cantidad de neuronas y capas, sumado a un diseño de arquitectura eficiente, debido a su complejidad la cantidad de características requeridas en la primera capa es mayor a la cantidad de características que posee nuestro conjunto de datos de entrenamiento (CD11) que es de 44x28 pixeles, en el caso de GoogleNet esta requiere imágenes de dimensión 224x224 pixeles y AlexNet solicita imágenes de dimensión 227x227 pixeles, la gran dimensión de las imágenes de entrada a su primera capa pueden adecuarse a su arquitectura de gran cantidad de capas y neuronas. La complejidad de la arquitectura de las CNN estándar implica también mayor tiempo computacional en los pasos *feedforward, backward y update*.

En cuanto a nuestra propuesta, los resultados presentados en la tabla V demuestran que un modelo CNN es preferido para el reconocimiento de automóviles, a comparación de C-Haar y SVM. Como se mencionó en la sección V-A, CD21 representa el conjunto de datos extraídos de una escena con escasa iluminación, Fig. 2c, el porcentaje obtenido por CNN-M11 es de 88 % que supera los porcentajes obtenidos por C-Haar y SVM con CD21. Para el caso de CD22, Fig. 2d, las muestras son obtenidas en una escena nocturna, donde los faros de los automóviles dificultan parcialmente la visión del mismo, el modelo CNN-M11 obtiene menor precisión que con C-Haar, aún así los porcentajes son bajos y se espera mejorar el reconocimiento de automóviles en escenas nocturnas para conseguir una mayor precisión.

VI. CONCLUSIONES

En el presente trabajo de investigación se ha realizado tres experimentos. El primer experimento tuvo como finalidad probar la eficiencia del modelo LeNet-5 frente a nuestro conjunto de datos, las predicciones de LeNet-5 no fueron satisfactorios en comparación con su rendimiento en MNIST, pero si se pudo observar la influencia de la media en el reconocimiento de automóviles. En cuanto al tamaño del parámetro batch no existe una regla para usar un determinado valor, este depende

Tabla VI: Subconjunto de imágenes tipo Caltech.

Muestras positivas	Muestras negativas

de la arquitectura y del número de iteraciones que se va a realizar en el entrenamiento.

El segundo experimento tuvo como objetivo encontrar la mejor combinación de hiperparámetros y el análisis de la influencia del tamaño del kernel en las convoluciones. ReLU, Xavier y Max predominaron en el mejor modelo de CNN para nuestro conjunto de datos, comprobando que su combinación es la más robusta para el reconocimiento de automóviles con CNN. Para el tercer experimento podemos resaltar que el conjunto de datos utilizados para el entrenamiento se diferencian del conjunto de datos de pruebas, CD21 y CD22, en la intensidad de iluminación, debido a que los datos de entrenamiento fueron capturados en horas con mayor iluminación, a diferencia de los datos de pruebas que fueron capturados en horas con menor iluminación. Los resultados demostraron que el modelo CNN denominado en este trabajo como M11 supera en precisión a la cascada de Haar y a SVM, aún cuando las imágenes son extraídas de una cámara de videovigilancia, y tomando en cuenta que los automóviles son multiescala y multiperspectiva. Con un modelo de CNN de 5 capas se puede realizar una eficiente clasificación binaria para imágenes extraídas de una cámara de videovigilancia, considerando variados contextos de iluminación. Podemos concluir que el uso de CNN es buen paso para alcanzar un clasificador robusto ante contextos de iluminación variados bajo las características de imágenes extraídas de una cámara de videovigilancia.

VII. AGRADECIMIENTOS

El presente trabajo es respaldado por el financiamiento 009-2014-FONDECYT del Consejo Nacional de Ciencia, Tecnología e Innovación Tecnológica(CONCYTEC-PERU) y por la Universidad Católica San Pablo de Arequipa.

- [2] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, 2015.
- [3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.
- [4] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, 2015.
- [5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [6] Caltech, "www.vision.caltech.edu/html-files/archive.html," 1999.
- [7] X. Li, M. Ye, M. Fu, P. Xu, and T. Li, "Domain adaption of vehicle detector based on convolutional neural networks," *International Journal of Control, Automation and Systems*, pp. 1–12, 2015.
- [8] Y. Cai, X. Chen, H. Wang, and L. Chen, "Deep representation and stereo vision based vehicle detection," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on*, pp. 305–310, IEEE, 2015.
- [9] X. Song, T. Rui, Z. Zha, X. Wang, and H. Fang, "The adaboost algorithm for vehicle detection based on cnn features," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, p. 5, ACM, 2015.
- [10] ImageNet, "<http://www.image-net.org>," 2012.
- [11] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proceedings of the ACM International Conference on Multimedia*, pp. 675–678, ACM, 2014.
- [12] OpenCV, "<http://www.opencv.org/>," 2016.
- [13] Dahua-Technology, "<http://www.dahuasecurity.com/es/>," 2010.
- [14] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.
- [15] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [16] D. He, C. Lang, S. Feng, X. Du, and C. Zhang, "Vehicle detection and classification based on convolutional neural network," in *Proceedings of the 7th International Conference on Internet Multimedia Computing and Service*, p. 3, ACM, 2015.
- [17] Apple.com, "<https://developer.apple.com/library/ios/documentation/>," 2016.
- [18] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, pp. 249–256, 2010.
- [19] J. Nagi, F. Ducatelle, G. A. Di Caro, D. Cireşan, U. Meier, A. Giusti, F. Nagi, J. Schmidhuber, and L. M. Gambardella, "Max-pooling convolutional neural networks for vision-based hand gesture recognition," in *Signal and Image Processing Applications (ICSIPA), 2011 IEEE International Conference on*, pp. 342–347, IEEE, 2011.

REFERENCIAS

- [1] H. Wang, Y. Cai, and L. Chen, "A vehicle detection algorithm based on deep belief network," *The scientific world journal*, vol. 2014, 2014.