



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**Departamento:**

**Ingeniería e Investigaciones Tecnológicas**

**Programa de acreditación:  
CyTMA2**

**Código del Proyecto:  
C2-ING-083**

**Título del proyecto**

**Obtención de dataset a partir de imágenes capturadas para su utilización en sistema experto de reconocimiento de imágenes.**

**Director:**

**Lic. Maidana, Carlos Eduardo**

**Integrantes:**

**Ing. Ferreyra Biron, Martín**

**Lic. Fiter, Jorge**

**Ing. Rodríguez, Carlos Alberto**

**Resolución Rectoral de acreditación: N° 444/21**

**Fecha de inicio: 01/01/2021**

**Fecha de finalización: 31/12/2022**



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

## **A. Desarrollo del proyecto (adjuntar el protocolo)**

**A.1.** Grado de ejecución de los objetivos inicialmente planteados, modificaciones o ampliaciones u obstáculos encontrados para su realización (desarrolle en no más de dos (2) páginas)

### **Análisis de tareas realizadas y grado de cumplimiento**

Como se propuso en este proyecto de investigación, finalmente se recolectaron 100 horas de videos, en distintas condiciones climáticas, de iluminación, con distintas cámaras y en distintas zonas geográficas, predominantemente en el área de Morón, Ituzaingó, Villa Luzuriaga, Ramos Mejía, Lomas del Mirador, San Justo y algunas zonas de CABA. Habiéndose finalizado con este procedimiento, reacondicionamos algunos programas para la recolección de datos y continuamos con la construcción del dataset propuesto.

Finalmente obtuvimos un dataset de 75 GBytes, que queda a disposición de la universidad (ya sea como material de estudio o material para próximas investigaciones) . Las cantidades de datos obtenidos fueron:

1. 29278 imágenes de vehículos.
2. 33829 patentes, de las cuales 11301 corresponden a la versión 2015 y 22528 a la versión 1994.
3. 62810 caracteres de patentes del tipo 1994, con una una distribución no uniforme y 62958 dígitos, del mismo tipo de patentes, dispuestos en una una distribución similar a una uniforme.
4. 42704 caracteres de patentes del tipo 2015, los cuales poseen una preponderancia en los caracteres A,B,C,D y E y 32028 dígitos de patentes del mismo tipo, los cuales se encuentran dispuestos de una forma similar a una distribución uniforme.

Cabe mencionar que no a todos los vehículos le corresponde una imagen de una patente en el dataset y viceversa, al igual que los caracteres y dígitos, no a todos los caracteres y dígitos les corresponde una imagen de una patente. Los datos presentados en este documento son el resultado del dataset depurado luego de haber sido revisado en cuantiosas oportunidades de forma manual, de diversas formas y con numerosas herramientas que fueron programadas para tal fin. También cabe mencionar que esta tarea representó mucho más tiempo que el estimado.

Ya con el dataset creado, nos dispusimos a entrenar varias redes neuronales YoloV3-Tiny, haciendo uso de los datos obtenidos, para cumplir los objetivos de:



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

1. Reconocer un vehículo en una escena
2. Localizar una patente en un vehículo y determinar a qué tipo de patente pertenece
3. Localizar los caracteres y dígitos en la patente

Y luego entrenar dos redes neuronales Xception para cumplir con la finalidad de:

1. Reconocer los dígitos de ambos tipos de patentes.
2. Reconocer los caracteres de ambos tipos de patentes

Los resultados del entrenamiento en general nos dejaron satisfechos con el siguiente desempeño en los tests:

1. Un 95% de efectividad en la localización de una patente y su clasificación en la imagen de un vehículo.
2. Un 79% de efectividad al localizar caracteres y dígitos en patentes de tipo 2015 (Mercosur).
3. Un 73,73% de efectividad al localizar caracteres y dígitos en patentes 1994 (Patentes con 3 letras y 3 números).
4. Un 95% de efectividad, aproximadamente, en el reconocimiento de los caracteres en general.
5. Un 97% de efectividad, en el reconocimiento de los dígitos en general
6. Un 90% de efectividad, en la detección de vehículos en escena.

Los pesos y las configuraciones de estas redes neuronales quedarán a disposición de la universidad. Fruto de estos resultados, el trabajo “Confección de dataset de patentes de automóviles argentinos para entrenamiento de redes neuronales” fue presentado al Workshop de Investigadores en Ciencias de la Computación 2023.

Por otra parte se analizó la posibilidad de implementar el pipeline propuesto en un sistema embebido, llegando a la siguiente conclusión: si no existe hardware dedicado al procesamiento masivo, que es necesario para poder realizar las operaciones que precisan las redes neuronales, es poco factible poder cumplir con las tareas propuestas en el pipeline en un tiempo razonable. Es por este motivo que no nos fue posible implementarlo en una placa Raspberry Pi, aunque quizás, hubiera sido posible utilizando alguna VPU externa como el Neural Compute Stick de Intel.

Igualmente confeccionamos un pequeño sistema de prueba del tipo cliente-servidor, en el cual el servidor es ejecutado en un sistema embebido NvidiaJetson TK1 (el cual posee una GPU con una limitada capacidad), donde se reciben peticiones para localizar un automóvil en una escena, localizar una patente en una imagen de un automóvil y localizar los caracteres y dígitos en una patente. Estas peticiones son enviadas desde un cliente que se ejecuta en una computadora convencional que dirige el pipeline. Esta placa puede



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

ejecutar casi todo el pipeline en un tiempo razonable, menos la detección de los caracteres, que debe ser ejecutado en el cliente debido a las dimensiones que posee Xception.

Debido a la cantidad de tiempo que se consumió para poder entrenar correctamente las 6 redes neuronales mencionadas, y el extenso trabajo que significó las distintas revisiones del dataset, no pudimos llevar a cabo completamente la implementación de otras dos redes neuronales y su comparación, ni tampoco pudimos confeccionar el informe detallado sobre cada una de ellas. Es importante recordar que durante 2021 aún estaba vigente el aislamiento social preventivo obligatorio, producto de la pandemia de COVID-19.

En el anexo III se encuentran los resultados preliminares (informe de avance) y los resultados finales del presente proyecto.

## B. Principales resultados de la investigación

### B.1. Publicaciones en revistas (informar cada producción por separado)

Artículo 1:	
Autores	
Título del artículo	
N° de fascículo	
N° de Volumen	
Revista	
Año	
Institución editora de la revista	
País de procedencia de institución editora	
Arbitraje	Elija un elemento.
ISSN:	
URL de descarga del artículo	
N° DOI	

### B.2. Libros

Libro 1	
Autores	
Título del Libro	
Año	
Editorial	
Lugar de impresión	
Arbitraje	Elija un elemento.
ISBN:	
URL de descarga del libro	
N° DOI	



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

### B.3. Capítulos de libros

Autores	
Título del Capítulo	
Título del Libro	
Año	
Editores del libro/Compiladores	
Lugar de impresión	
Arbitraje	Elija un elemento.
ISBN:	
URL de descarga del capítulo	
N° DOI	

### B.4. Trabajos presentados a congresos y/o seminarios

Autores	<i>Ferreyra Biron Martín; Maidana, Carlos Eduardo</i>
Título	<i>Confección de dataset de patentes de automóviles argentinos, para el entrenamiento de redes neuronales</i>
Año	<i>2023</i>
Evento	<i>WICC 2023</i>
Lugar de realización	<i>UNNOBA</i>
Fecha de presentación de la ponencia	<i>13 y 14 de abril de 2023</i>
Entidad que organiza	<i>UNNOBA</i>
URL de descarga del trabajo (especificar solo si es la descarga del trabajo; formatos pdf, e-pub, etc.)	<i><a href="https://wicc2023.unnoba.edu.ar/2023/04/12/confeccion-de-dataset-de-patentes-de-automoviles-argentinos-para-entrenamiento-de-redes-neuronales/">https://wicc2023.unnoba.edu.ar/2023/04/12/confeccion-de-dataset-de-patentes-de-automoviles-argentinos-para-entrenamiento-de-redes-neuronales/</a></i>

### B.5. Otras publicaciones

Autores	
Año	
Título	
Medio de Publicación	



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**C. Otros resultados. Indicar aquellos resultados pasibles de ser protegidos a través de instrumentos de propiedad intelectual, como patentes, derechos de autor, derechos de obtentor, etc. y desarrollos que no pueden ser protegidos por instrumentos de propiedad intelectual, como las tecnologías organizacionales y otros. Complete un cuadro por cada uno de estos dos tipos de productos.**

C.1. Títulos de propiedad intelectual. Indicar: Tipo (marcas, patentes, modelos y diseños, la transferencia tecnológica) de desarrollo o producto, Titular, Fecha de solicitud, Fecha de otorgamiento

Tipo	Titular	Fecha de Solicitud	Fecha de Emisión

C.2. Otros desarrollos no pasibles de ser protegidos por títulos de propiedad intelectual. Indicar: Producto y Descripción.

Producto	Descripción

**D. Formación de recursos humanos. Trabajos finales de graduación, tesis de grado y posgrado. Completar un cuadro por cada uno de los trabajos generados en el marco del proyecto.**

D.1. Tesis de grado

Director (apellido y nombre)	Autor (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis

D.2 Trabajo Final de Especialización

Director (apellido y nombre)	Autor (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título del Trabajo Final

D.2. Tesis de posgrado: Maestría

Director (apellido y nombre)	Tesista (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

nombre)					

### D.3. Tesis de posgrado: Doctorado

Director (apellido y nombre)	Tesista (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título de la tesis

### D.4. Trabajos de Posdoctorado

Director (apellido y nombre)	Posdoctorando (apellido y nombre)	Institución	Calificación	Fecha /En curso	Título del trabajo	Publicación

### E. Otros recursos humanos en formación: estudiantes/ investigadores (grado/posgrado/ posdoctorado)

Apellido y nombre del Recurso Humano	Tipo	Institución	Período (desde/hasta)	Actividad asignada <sup>1</sup>

**F. Vinculación<sup>2</sup>:** Indicar conformación de redes, intercambio científico, etc. con otros grupos de investigación; con el ámbito productivo o con entidades públicas. Desarrolle en no más de dos (2) páginas.

<sup>1</sup> Descripción de la/s actividad/es a cargo (máximo 30 palabras)

<sup>2</sup>Entendemos por acciones de “vinculación” aquellas que tienen por objetivo dar respuesta a problemas, generando la creación de productos o servicios innovadores y confeccionados “a medida” de sus contrapartes.



<b>Código</b>	FPI-009
<b>Objeto</b>	Guía de elaboración de Informe final de proyecto
<b>Usuario</b>	Director de proyecto de investigación
<b>Autor</b>	Secretaría de Ciencia y Tecnología de la UNLaM
<b>Versión</b>	5
<b>Vigencia</b>	03/9/2019

**G. Otra información. Incluir toda otra información que se considere pertinente.**

**H. Cuerpo de anexos:**

- Anexo I: Copia de cada uno de los trabajos mencionados en los puntos B, C y D, y certificaciones cuando corresponda.<sup>3</sup>
- Anexo II:
  - FPI-013: Evaluación de alumnos integrantes. (si corresponde)
  - FPI-014: Comprobante de liquidación y rendición de viáticos. (si corresponde)
  - FPI-015: Rendición de gastos del proyecto de investigación acompañado de las hojas foliadas con los comprobantes de gastos.
  - FPI-035: Formulario de reasignación de fondos en Presupuesto.
- Nota justificando baja de integrantes del equipo de investigación.

Lic. Carlos Eduardo Maidana  
Director del proyecto.

Lugar y fecha: San Justo 24 de abril de 2023

- Cargar este formulario junto con los documentos correspondientes **exclusivamente** al Anexo I en SIGEVA UNLaM. Realizar la presentación impresa de los mismos junto con los restantes Anexos en la Secretaría de Investigación de la unidad académica correspondiente.

<sup>3</sup>En caso de libros, podrá presentarse una fotocopia de la primera hoja significativa o su equivalente y el índice.



# ANEXO I

# CONFECCIÓN DE DATASET DE PATENTES DE AUTOMÓVILES ARGENTINOS PARA ENTRENAMIENTO DE REDES NEURONALES.

Ferreira Biron, Martín, Maidana, Carlos Eduardo



Universidad Nacional de la Matanza

## 1. Contexto

Este trabajo es el resultado de la investigación "Obtención de dataset a partir de imágenes capturadas para su utilización en sistema experto de reconocimiento de imágenes" la cual es una continuación de una investigación anterior inconclusa por distintas desavenencias y la imposibilidad de llevarla a cabo debido a las diversas situaciones ocasionadas por la pandemia de COVID-19.

## 2. Línea de investigación

Al necesitar la UNLaM una forma de confirmar qué automóviles ingresaban a su estacionamiento se propuso entrenar un sistema experto con el fin de detectar las patentes de los vehículos. Pero nos encontramos con el inconveniente de no disponer de una base de datos de imágenes con patentes argentinas. Debido a esto llevamos a cabo esta investigación en la que se propuso crear un dataset de patentes argentinas y utilizarlo en el entrenamiento de diversas redes neuronales.

Para elaborar este dataset se planteó obtener al menos **100 horas de grabación** de tráfico a fin de revisarlas y capturar las imágenes de vehículos, patentes y caracteres para así crear el dataset, el cual contendría las posiciones de los elementos capturados en la imagen, su alto, su ancho y su tipo. Sumado a esto se conservarían los fotogramas donde se encontrarían cada una de las imágenes recolectadas.

Luego de comprobar los datos recolectados concluiríamos la creación del dataset y podríamos realizar los cálculos estadísticos pertinentes. Ya con el dataset creado y valiéndolos de él podríamos ser capaces de desarrollar un sistema que pudiera detectar, a partir del uso de redes neuronales, vehículos en una escena, luego su patente y finalmente los caracteres que la conforman. De esta forma comprobaríamos la idoneidad del dataset confeccionado.

Para cumplir dichos objetivos consideramos la red neuronal Yolo V3, en su versión tiny, para la detección de vehículos, localización de caracteres y patentes. Para reconocer los caracteres en cambio utilizamos la red neuronal Xception quedando el pipeline como se muestra a continuación:



## 3. Resultados obtenidos

El primer resultado a resaltar es la cantidad de **imágenes de vehículos** recolectados que asciende a la cifra de **29728**. Por otro lado la distribución de patentes caracteres y dígitos recolectados son las que se muestra a continuación:

Distribución de patentes (cantidad)



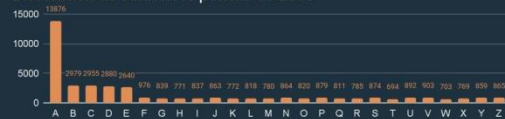
Distribución de caracteres de patente 1994



Distribución de dígitos de patente 1994



Distribución de caracteres patente de 2015



Distribución de dígitos patente de 2015



Con los datos obtenidos, entrenamos las redes neuronales que se mencionan obteniéndose resultados positivos mayormente. Para el caso de la detección de vehículos utilizamos una red Yolo V3-Tiny pre entrenada. La misma fue entrenada por 500.000 epochs utilizando el 70% de las imágenes de los vehículos obtenidas para entrenamiento y el restante para testeo. Así obtuvimos un **92% de efectividad en la detección de vehículos sobre el total de imágenes de los mismos.**

Siguiendo un procedimiento similar entrenamos otra red neuronal Yolo V3-Tiny para localizar en la imagen de un vehículo la patente del mismo y categorizarla (detectar si corresponde a la versión 1994 o 2015). En esta red obtuvimos un **porcentaje de acierto del 95%** sobre el total de patentes con vehículos del dataset.

Para localizar en una patente los caracteres y dígitos utilizamos dos redes neuronales Yolo V3-Tiny, una para localizar los caracteres y dígitos en las patentes 1994 y otra especializada para realizar la misma tarea pero en patentes 2015. En este caso entrenamos ambas redes con un millón de epochs aproximadamente y dividimos en cada caso el conjunto de patentes en 3: un 10% para validación, del 90% restante un 70% para entrenamiento y el resto para testing. Entrenamos la red creando nuevos conjuntos aleatorios de entrenamiento y de testing (siempre sin contar al conjunto de validación) cada 50.000 epochs y obtuvimos un **porcentaje de acierto del 79%** en el caso de patentes 2015 y **73,73%** en el caso de patentes 1994 en el conjunto de datos de validación. Cabe aclarar que la forma en cómo se tomaron estas mediciones involucraron también una métrica de IOU, la cual si es relajada impacta sensiblemente y positivamente en el porcentaje de acierto de la localización de caracteres y dígitos.

Finalmente, y con la misma técnica descrita anteriormente se realizó el entrenamiento de las dos redes neuronales Xception (pre entrenadas) mencionadas en el pipeline. En el caso del entrenamiento de la red neuronal destinada a reconocer caracteres, descartamos las últimas capas de la red y agregamos tres MLPs de 800 perceptrones. Entrenamos las capas agregadas y las dos últimas convolucionales durante 40 epochs **obteniéndose un porcentaje de acierto en el conjunto de datos de validación del 95,11%.**

En el caso del entrenamiento de la red neuronal destinada a reconocer dígitos, procedimos de la misma forma que en el caso anterior, aunque en este solamente entrenamos las MLPs agregadas y la última capa convolucional durante 120 epochs **obteniéndose así un porcentaje de acierto del 97,79%** en el conjunto de validación.

El dataset obtenido queda a disposición de la UNLaM, no siendo de dominio público y si desea ser obtenido para utilizarlo en investigaciones, se necesitará primero la autorización de la universidad.

## 4. Formación de recursos humanos

Los integrantes de este desarrollo forman parte del plantel docente e investigadores del Departamento de Ingeniería e Investigaciones Tecnológicas de la Universidad Nacional de la Matanza. En conjunto, los integrantes de este grupo poseen conocimientos de electrónica e informática.

## **ANEXO III**

## Resultados preliminares obtenidos (año 2021)

Al comenzar a realizar las grabaciones y al tener a nuestra disposición nuevos vídeos, realizamos nuevas modificaciones en el programa segmentador para agilizar el proceso de segmentación manual. Luego de varios videos y horas de segmentación, nos percatamos de que este proceso es realmente lento de realizar, incluso con las herramientas ya desarrolladas en la investigación anterior, por lo tanto realizamos nuevas modificaciones para agilizar aún más la segmentación. Debido a la cantidad de horas necesarias para realizar la segmentación manual nos propusimos alcanzar el número de 5000 patentes, considerando que estas serán suficientes para poder realizar las pruebas para poder entrenar y comparar las distintas redes neurales a estudiar. Este número fue alcanzado satisfactoriamente y a continuación se mostrará un análisis de los datos obtenidos.

Cada uno de los elementos segmentados en estos videos fueron guardados en una carpeta, junto con el fotograma de la escena correspondiente (esto será útil en el caso de redes neuronales convolucionales como YOLO). Esta carpeta se encuentra acompañada por un archivo XML que estará compuesto por los siguientes elementos.

### Automóviles:

```
<objeto class="org.unlam.structure.Auto">
<filename>1617750490978.jpg</filename>
<position>
<x>979</x>
<y>538</y>
</position>
<dimension>
<width>225</width>
<height>198</height>
</dimension>
<scene>2019_09_21_205356.MOV_851.png</scene>
<night>true</night>
</objeto>
<tipo>AUTO</tipo>
</org.unlam.structure.ObjetoXML>
```

## Patentes:

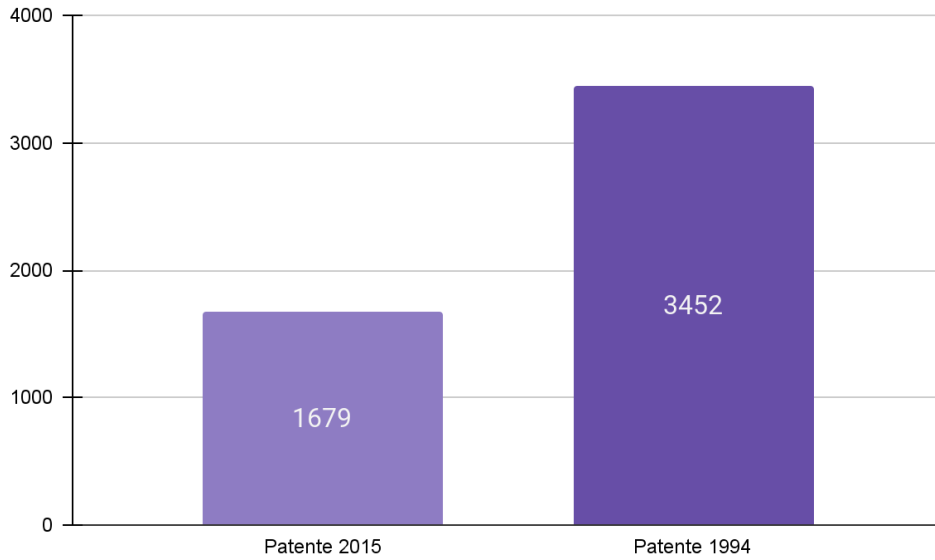
```
<org.unlam.structure.ObjetoXML>  
<objeto class="org.unlam.structure.Patente">  
<filename>1617750656558.jpg</filename>  
<patenteletras>JQK077</patenteletras>  
<patentenueva>>false</patentenueva>  
<position>  
<x>863</x>  
<y>653</y>  
</position>  
<dimension>  
<width>56</width>  
<height>24</height>  
</dimension>  
<scene>2019_09_21_205356.MOV_146523.png</scene>  
<night>>true</night>  
</objeto>  
<tipo>PATENTE</tipo>  
</org.unlam.structure.ObjetoXML>
```

## Caracteres:

```
<org.unlam.structure.ObjetoXML>
<objeto class="org.unlam.structure.Caracteres">
<caracteres>
<org.unlam.structure.InformacionCaracteres>
<filename>1617750668256.jpg</filename>
<caracter>j</caracter>
<patentenueva>>false</patentenueva>
<posicion>
<x>8</x>
<y>8</y>
</posicion>
<dimension>
<width>7</width>
<height>10</height>
</dimension>
</org.unlam.structure.InformacionCaracteres>
<org.unlam.structure.InformacionCaracteres>
<filename>1617750672991.jpg</filename>
<caracter>q</caracter>
<patentenueva>>false</patentenueva>
<posicion>
<x>14</x>
<y>8</y>
</posicion>
<dimension>
<width>6</width>
<height>10</height>
</dimension>
</org.unlam.structure.InformacionCaracteres>
<org.unlam.structure.InformacionCaracteres>
<filename>1617750675655.jpg</filename>
<caracter>k</caracter>
<patentenueva>>false</patentenueva>
<posicion>
<x>19</x>
<y>7</y>
</posicion>
<dimension>
<width>7</width>
<height>10</height>
</dimension>
</org.unlam.structure.InformacionCaracteres>
<org.unlam.structure.InformacionCaracteres>
<filename>1617750682313.jpg</filename>
<caracter>0</caracter>
<patentenueva>>false</patentenueva>
<posicion>
<x>30</x>
<y>7</y>
</posicion>
<dimension>
<width>7</width>
<height>10</height>
</dimension>
</org.unlam.structure.InformacionCaracteres>
<org.unlam.structure.InformacionCaracteres>
<filename>1617750685704.jpg</filename>
<caracter>7</caracter>
<patentenueva>>false</patentenueva>
<posicion>
<x>37</x>
<y>7</y>
</posicion>
<dimension>
<width>7</width>
<height>11</height>
</dimension>
</org.unlam.structure.InformacionCaracteres>
<org.unlam.structure.InformacionCaracteres>
<filename>1617750689344.jpg</filename>
<caracter>7</caracter>
<patentenueva>>false</patentenueva>
<posicion>
<x>43</x>
<y>7</y>
```

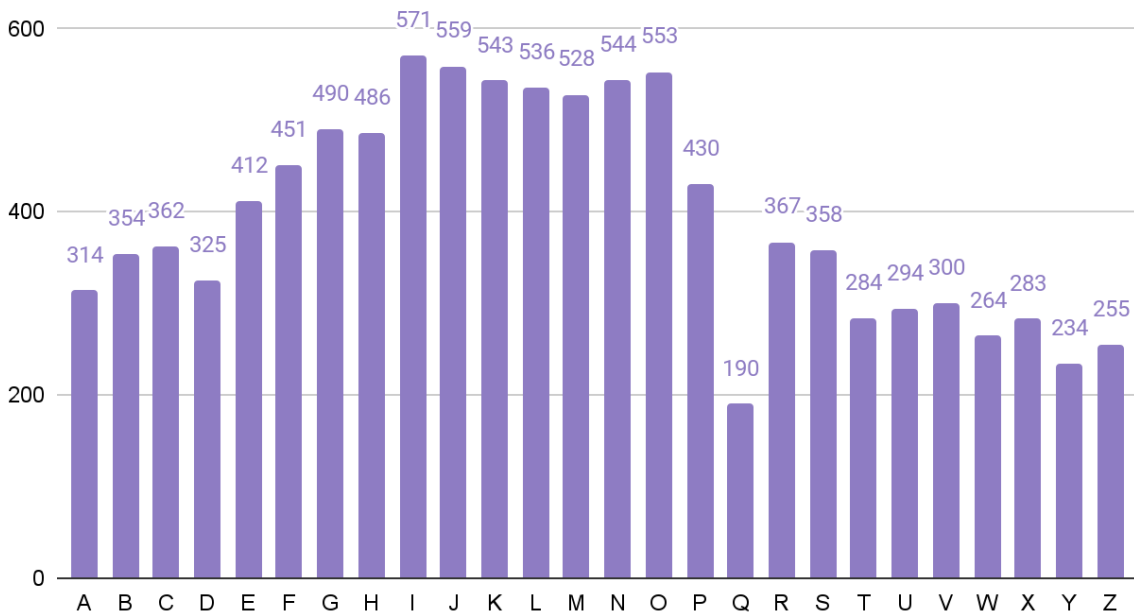
Nótese que se guardan las posiciones de los automóviles, patentes respecto a la escena, y la posición de los caracteres respecto a la patente, pudiendo así recuperar los elementos segmentados.

De 6356 autos se obtuvieron 5131 patentes que se encuentran distribuidas de la siguiente forma:



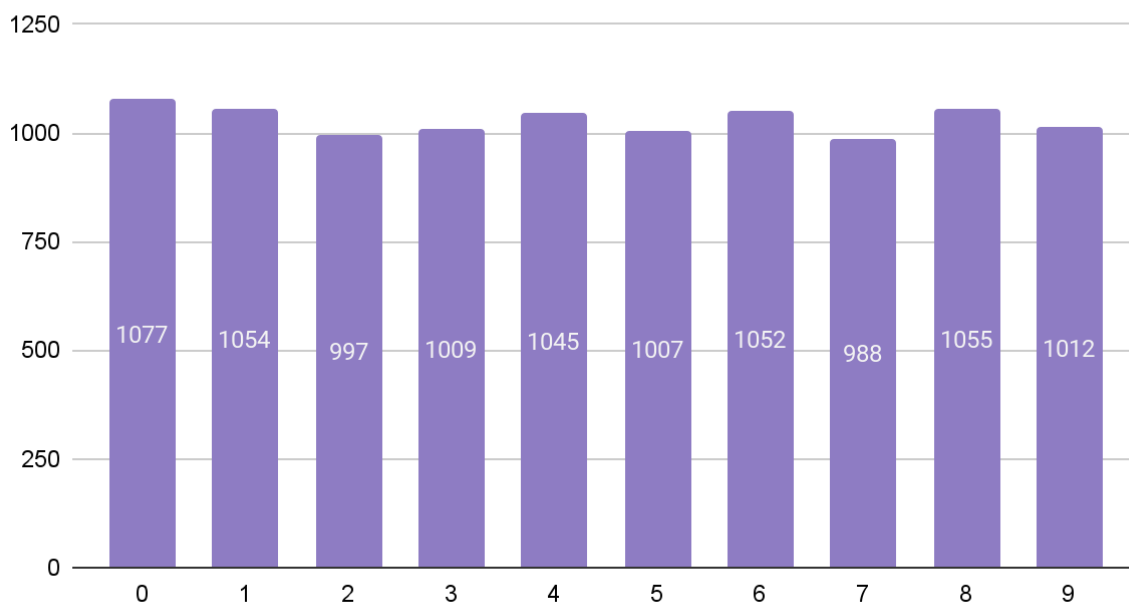
Cómo se puede observar la cantidad de patentes 1994 supera por un poco más del doble a la cantidad de patentes 2015 obtenidas. Por otra parte se recolectaron 10287 caracteres que se encuentran asignados como se muestran a continuación

### Distribución de caracteres de patente 1994



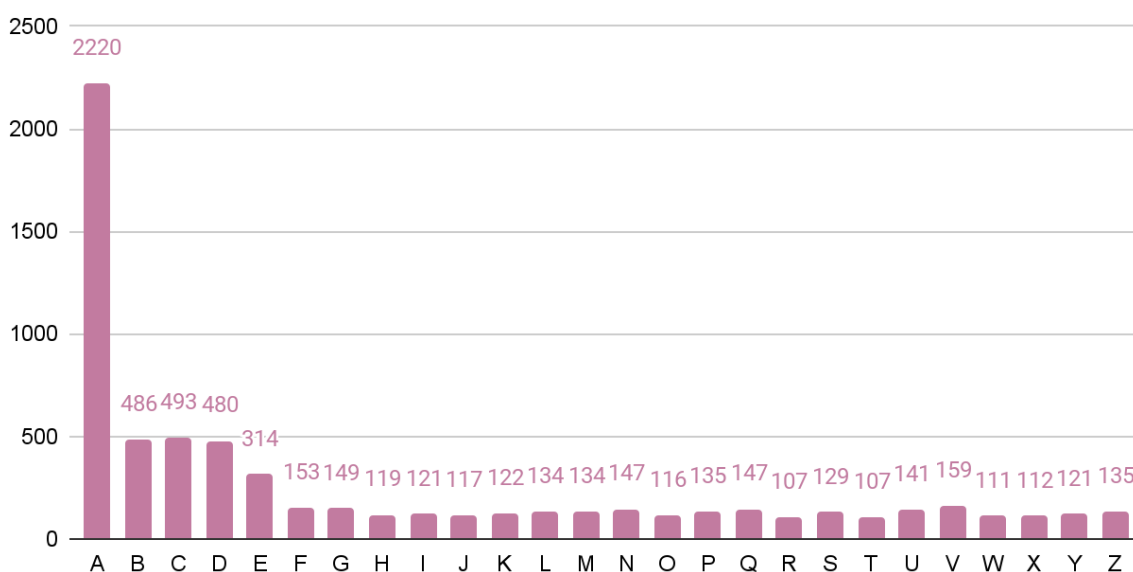
La distribución no se asemeja a una uniforme, sin embargo esto radica principalmente en cómo las patentes fueron distribuidas a lo largo de los años 1994-2015, por lo tanto es natural observar que los caracteres I,J,K,L,M N y O sean los caracteres con más apariciones.

### Distribución de dígitos de patente 1994



Respecto a la patentes 1994, se pudieron recolectar 10296 dígitos, cuya distribución se puede observar en el gráfico. A diferencia de los caracteres, se puede observar como es esperable, una distribución mucho más uniforme.

### Distribución de caracteres patente de 2015

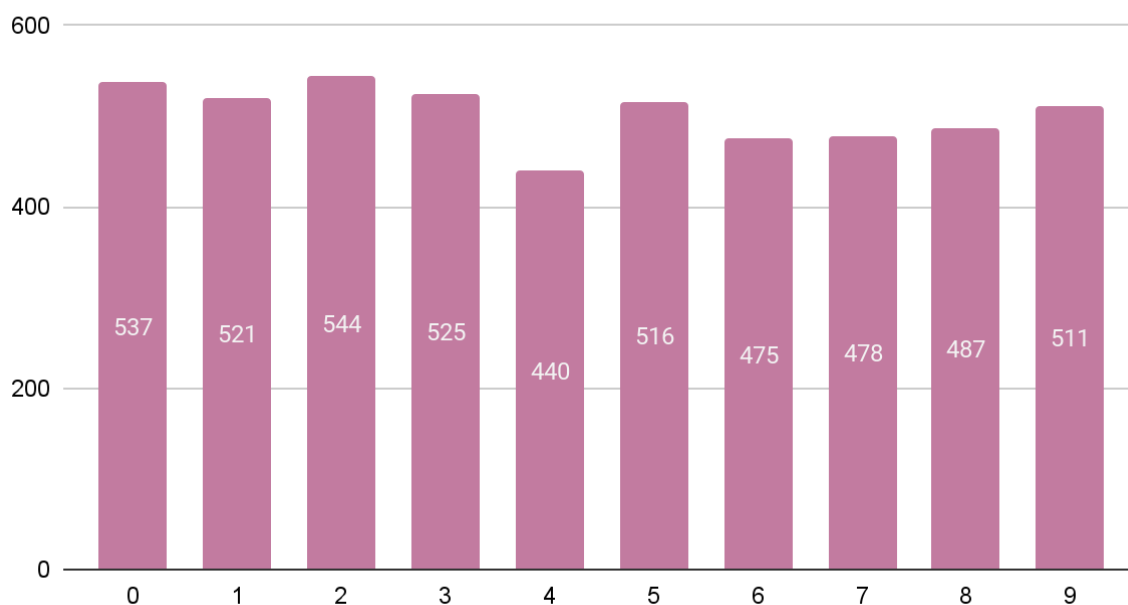


Respecto a las patentes 2015, pudimos reunir 6709 caracteres, distribuidos como se observa en el gráfico. En este caso como es esperable la cantidad de letras A será mucho más alta que los



demás caracteres debido a la forma en cómo se distribuyen dichas patentes. Esta es la razón por la cual los caracteres B, C, D y E los continúan en cantidad de apariciones. Desde la F en adelante el comportamiento de la cantidad de caracteres se asemeja a una distribución uniforme. Lamentablemente creemos, en estos casos, que la cantidad de caracteres obtenidas no es todavía suficiente.

## Distribución de dígitos patente de 2015



Con esta fracción del dataset pudimos obtener 5034 dígitos de patentes 2015 distribuidos como se muestra en el gráfico. Al igual que en el caso de los dígitos de patentes 1994, la distribución de los mismos se asemeja a una distribución uniforme.

Sumado a lo anterior, continuamos la investigación comenzando un curso de inteligencia artificial enfocado en la utilización de Python y Keras seguimos el estudio sobre este campo con otro curso enfocado específicamente en el uso de redes neuronales convolucionales. Con la información que obtuvimos en los mismos pudimos ver en cierto punto el estado del arte y decidimos estudiar las redes neuronales Yolo (en su versión 3 o posterior), Mobilenet (en su versión 2 o 3) y PeleeNet las cuales en teoría no necesitaran un tiempo considerable en su entrenamiento como otras redes neuronales convolucionales demasiado profundas, cuyo entrenamiento se encuentra en el orden de los días cuando se tratan de datasets realmente voluminosos.

Paralelamente al llegar a la cantidad de patentes obtenidas mostradas en este informe, comenzamos el entrenamiento a hacer la primera prueba de la red Yolo V3. Esto lo realizamos para incrementar con la etapa de la búsqueda de datos más rápidamente. Para esto tomamos todo el conjunto de dígitos de la patente de 1994, realizamos un proceso de data argumentation principalmente en lo que respecta a la iluminación y la utilizamos como entrada de una red neuronal Yolo V3 tiny. Luego integramos esta red neuronal con el programa desarrollado para crear el data set, de modo tal que cada vez que se seleccionara una patente, la red neuronal reconociera los caracteres de la misma. Si bien el entrenamiento fue exitoso y la tasa

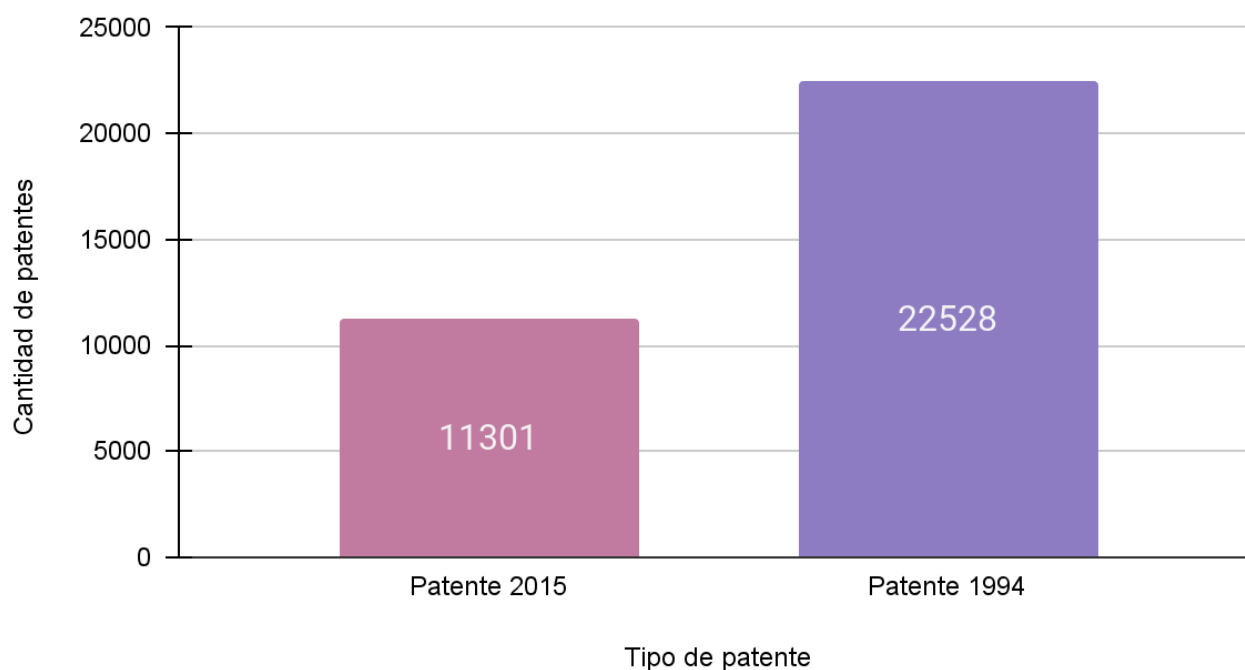
de aciertos en el lote de testing comenzaba a sobrepasar el 90%, no era en la generalidad de los casos en donde la red neuronal funcionaba de forma correcta, sin embargo creemos que esto se debe al dataset (el cual seguimos incrementando) o a una mala configuración de la red.

## Resultados obtenidos al finalizar el proyecto (2022)

Luego de una extensa recolección de patentes, autos y caracteres de patentes (que nos ocupó mucho más tiempo y trabajo del planificado) creamos un dataset que contiene las clases mencionadas. Recordemos que el dataset está compuesto por imágenes capturadas a lo largo de 100 horas de grabación de video. Luego de que el mismo fue depurado varias veces, obtuvimos los siguientes resultados.

La cantidad de imágenes de vehículos obtenidas fue de 29278. De la mayoría de estas imágenes (debido a que no todas las patentes tienen relacionado un vehículo en este dataset) obtuvimos la siguiente distribución de patentes.

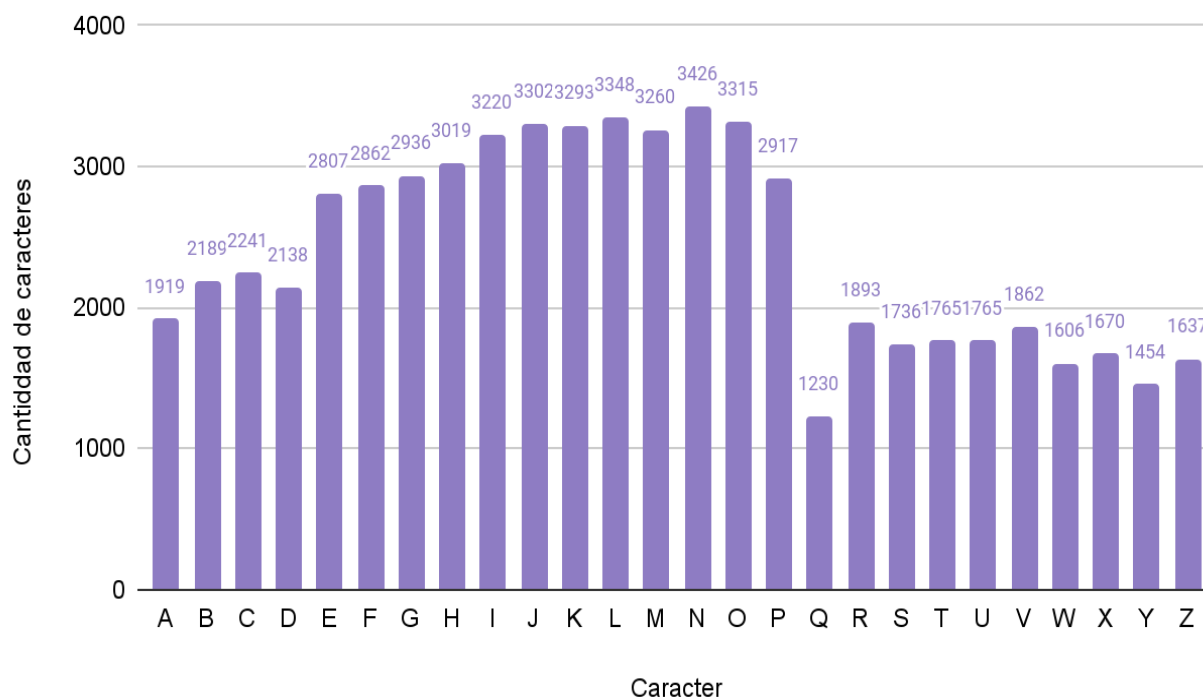
### Distribución de patentes



Se puede observar que la cantidad de patentes tipo 1994 duplica, aproximadamente, a las patentes de tipo 2015. Esta disparidad que obtuvimos es atribuible a la distribución geográfica de nuevos rodados con nuevas patentes y con las zonas en la cual fueron grabados los videos. Si bien se intentó que las horas de grabación, fueran lo más heterogéneas posibles, adquiriendo estos videos en distintos puntos del Gran Buenos Aires y ciertas zonas de Capital Federal, los videos que fueron grabados en la zona de Gran Buenos Aires tuvieron mayor preponderancia.

Por otra parte, la cantidad de caracteres correspondientes a las patentes 1994 obtenidas fueron 62810 cuya distribución es la que se muestra a continuación.

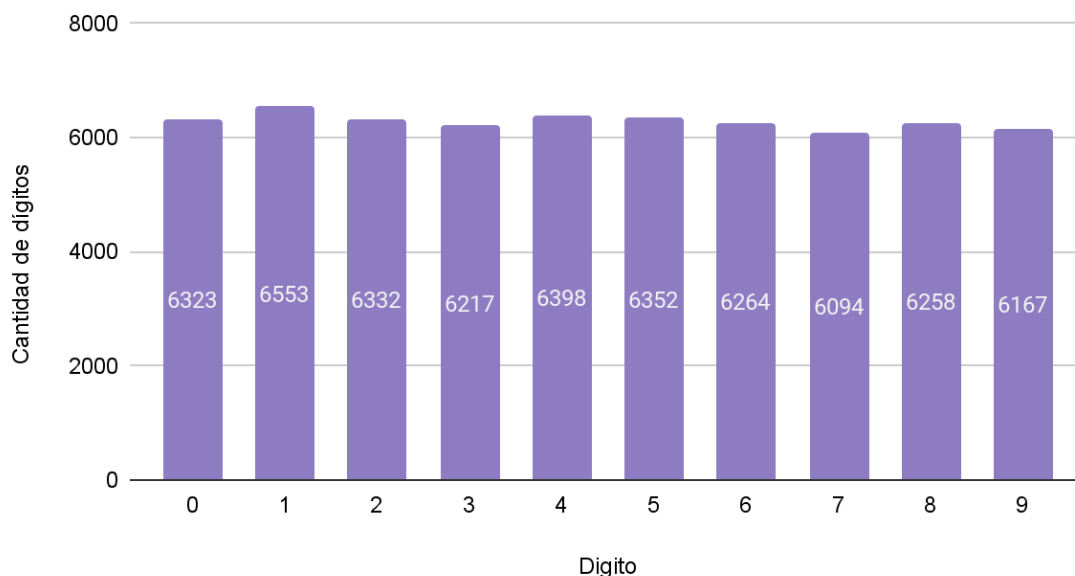
## Distribución de caracteres de patente 1994



Se puede contemplar que la disposición de los caracteres no se aproxima a una distribución uniforme. Esto se hizo notar en el informe de avance de 2021, pero lo consideramos algo esperable debido a como fueron distribuidas las patentes a lo largo de los años. Existen más posibilidades de observar patentes que comienzan con K,L,M y N, que con A,B o C debido a que las del primer grupo corresponden a autos que poseen patentamientos más modernos y por lo tanto más numerosos en la actualidad. En cambio las del segundo grupo son patentes atribuidas a vehículos más antiguos, lo que hace más probable que sean rodados fuera de circulación en este momento. Algo similar ocurre con las letras V,W,X,Y y Z.

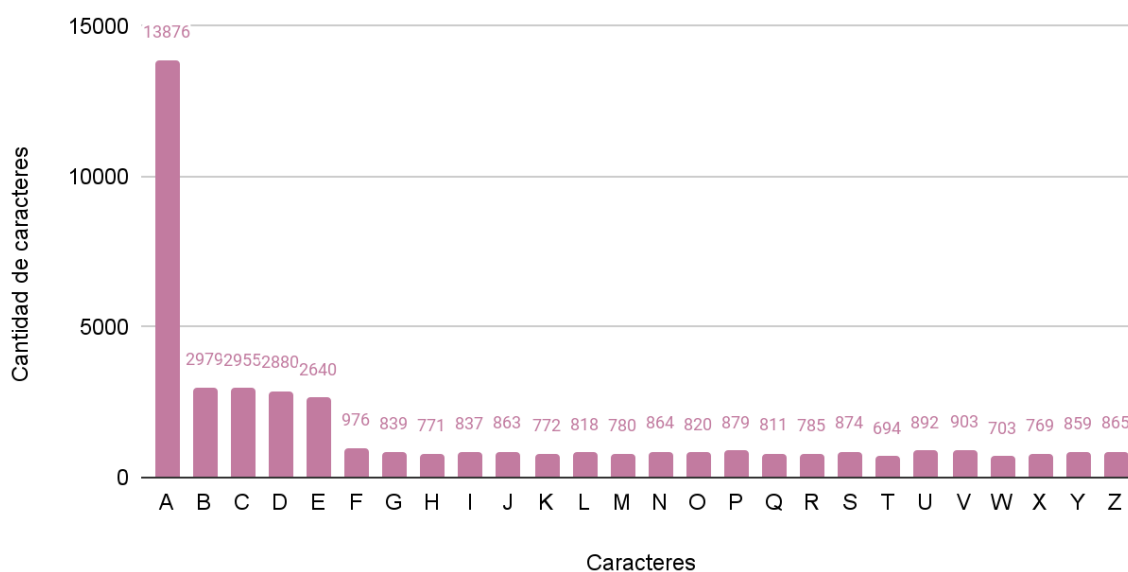
Distinto es el caso de los dígitos correspondientes a las patentes 1994. La cantidad de dígitos recolectados fue de 62958 cuya disposición se asemeja a una distribución uniforme. Esto es esperable debido a que los números fueron distribuidos aleatoriamente a lo largo de los años y no existe ningún tipo de relación entre los dígitos y el momento en que se emitió la patente, más allá de su natural orden secuencial. La distribución de los dígitos se puede apreciar en el gráfico que se muestra a continuación.

## Distribución de dígitos de patente 1994



Respecto a las patentes 2015 la cantidad de caracteres recolectados fue de 42704, cuya distribución se puede observar en el gráfico.

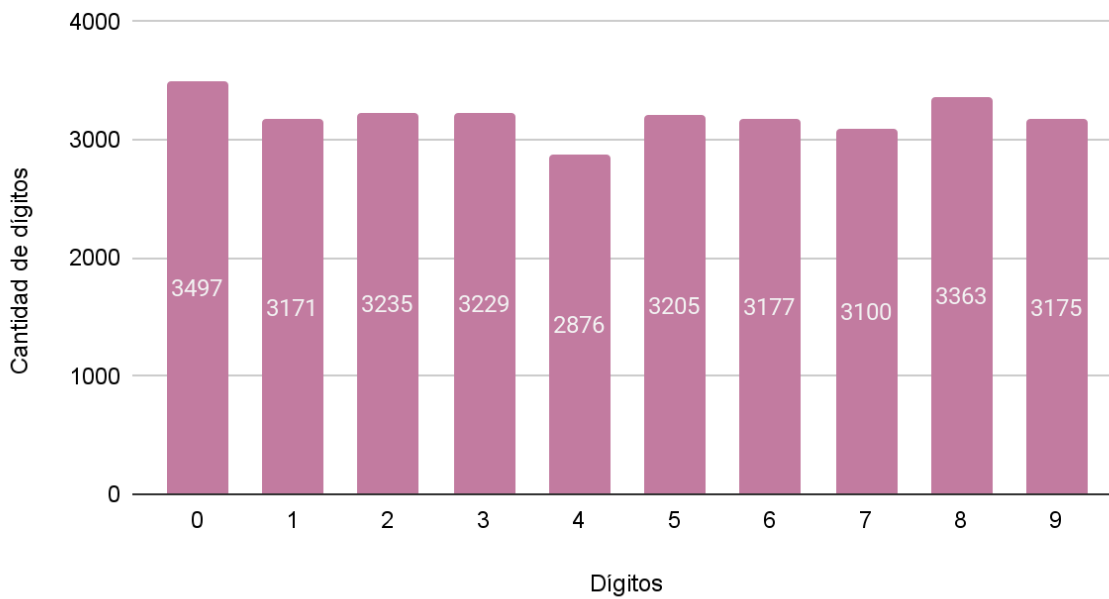
## Distribución de caracteres patente de 2015



Al igual que en el informe de avance, podemos advertir que la cantidad de caracteres posee una disposición similar a una distribución uniforme en el segmento F-Z, pero no ocurre lo mismo si analizamos el segmento de caracteres A-E. Debido a que todas las patentes 2015 recolectadas hasta el momento comienzan con una A, se puede observar una desproporcionada preponderancia de esta letra y en el segmento B-E se puede ver una cierta uniformidad (no compartida con el segmento de caracteres F-Z) debido a que son las que segundas letras más utilizadas en este tipo de patentes. Por otra parte, la cantidad de dígitos de patentes 2015 obtenidos fueron de 32028 cuya distribución se puede apreciar en el gráfico que se muestra a

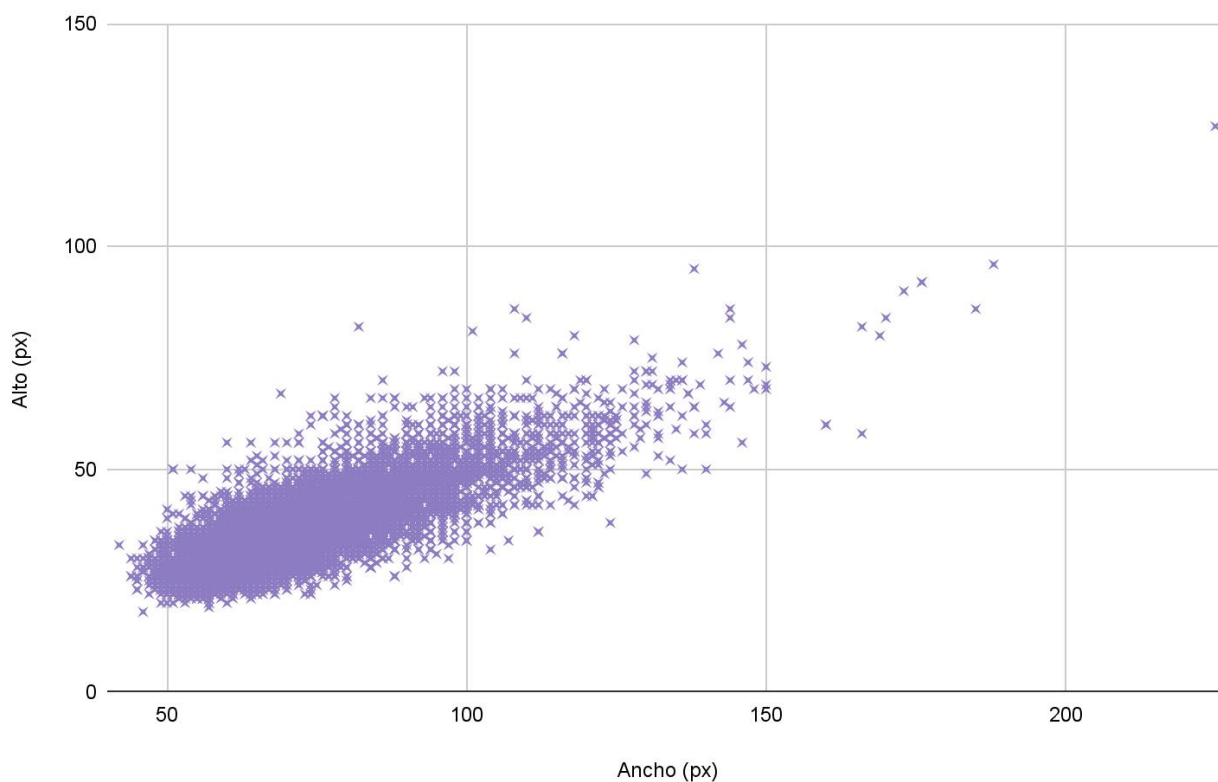
continuación.

### Distribución de dígitos patente de 2015

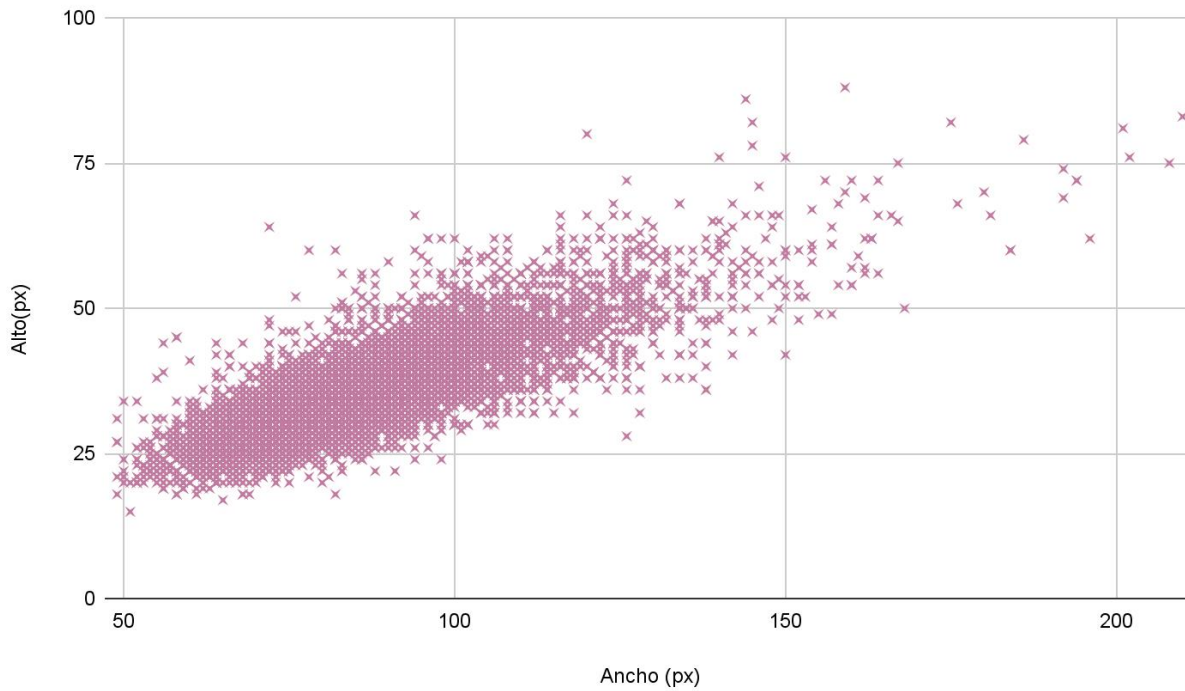


Al igual que en el caso de las patentes 1994 observamos a simple vista una distribución similar a una uniforme en los dígitos. Por otra parte, a continuación se muestra la dispersión de dimensiones de las patentes y los vehículos recolectados. Estos gráficos dan una visión del grado de uniformidad del dataset respecto a las dimensiones.

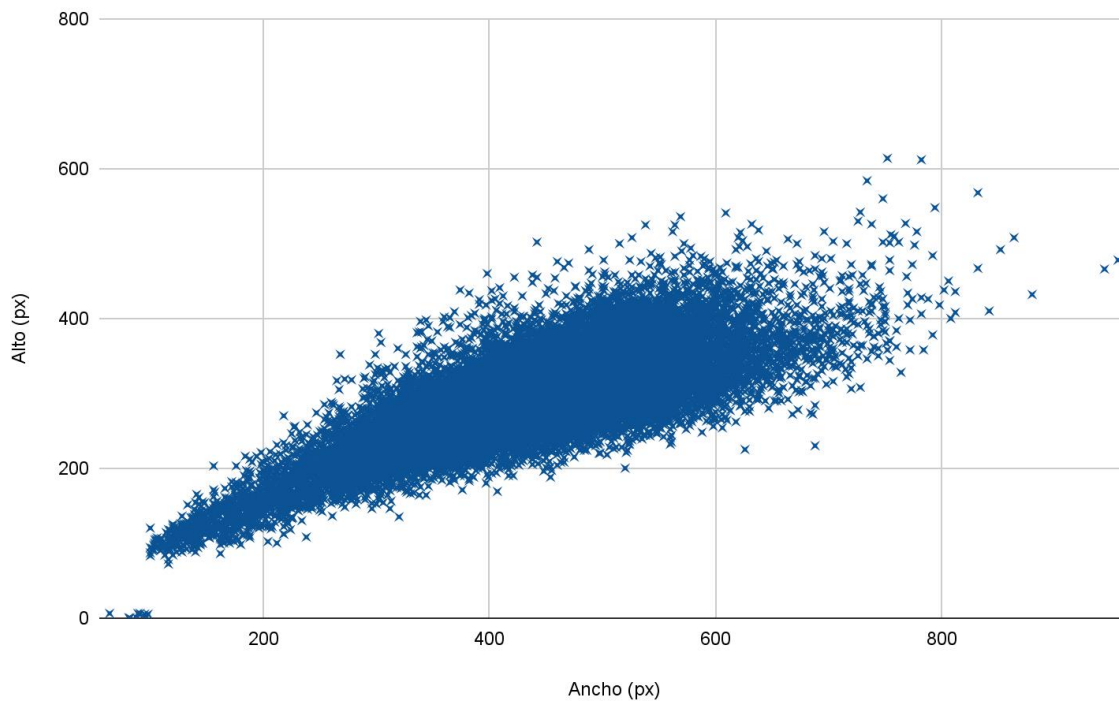
### Dispersión de dimensión (Patentes 1994)



### Dispersión de dimensión (Patentes 2015)



### Dispersión de dimensión (Vehículos)



Es importante notar que dimos por descartado el campo “noche” en todas las clases. Esto es debido a que las cámaras con las cuales fueron grabados los videos, no poseían la suficiente sensibilidad para distinguir los distintos caracteres de las patentes en pobres condiciones lumínicas. Como la gran mayoría de los videos fueron grabados por la mañana y por la tarde,

decidimos descartar dicho campo y agregar dos nuevos. Estos campos fueron añadidos a la clase patente y son el archivo del vehículo donde aparece la patente y la posición de esta respecto a la imagen del vehículo. Para lograr esto utilizamos un pequeño programa para detectar la posición de la patente haciendo uso de *match template*. Estos nuevos campos (agregados de manera automática) fueron validados de forma manual a posteriori.

Por último para que el uso de los datos sea más simple, decidimos desistir del formato XML y emplear distintos archivos CSV, dependiendo de la clase en cuestión. A continuación detallamos el formato de cada archivo CSV que conformará el dataset.

Para el caso de los vehículos el formato del archivo CSV es el siguiente:

```
filename,x,y,w,h,scene
000000.jpg,979,538,225,198,000001.jpg
000002.jpg,620,584,139,138,000003.jpg
000004.jpg,1054,567,130,116,000003.jpg
000005.jpg,435,573,214,164,000006.jpg
000007.jpg,1106,558,205,154,000008.jpg
...
```

Donde **filename** corresponde al nombre del archivo de la imagen del vehículo capturado, **x** e **y** son las coordenadas correspondientes al vértice superior izquierdo del vehículo respecto a la escena, **w** y **h** corresponden al ancho y al alto de la imagen capturada y finalmente, el campo **scene** corresponde al fotograma donde se encontró al vehículo.

Para el caso de los caracteres el formato es el siguiente:

```
character,filename,x,y,w,h,new_lp,lp_filename
J,000029.jpg,12,12,10,16,false,000028.jpg
Q,000030.jpg,22,11,9,17,false,000028.jpg
K,000031.jpg,30,11,11,17,false,000028.jpg
0,000032.jpg,48,11,10,17,false,000028.jpg
7,000033.jpg,57,10,9,17,false,000028.jpg
...
```

Donde **caracteres** el carácter, **filename** es el nombre del archivo de la imagen del carácter, **x** e **y** son las coordenadas correspondientes al vértice superior izquierdo del carácter respecto a la patente, **w** y **h** corresponden al ancho y al alto de la imagen capturada del carácter, **new\_lp** indica si el carácter corresponde a la patente 1994 o 2015 y por último **lp\_filename** hace referencia al nombre del archivo de la patente donde se encuentra el carácter.

Por último podemos observar el caso de las patentes, cuyo formato es el que se presenta a continuación.



```
filename,x,y,w,h,car_filename,x_car,y_car,new_lp,scene,characters
000028.jpg,771,704,89,39,000026.jpg,135,135,false,000027.jpg,jqk077
000072.jpg,838,746,110,46,000071.jpg,184,182,false,000070.jpg,jqk077
000294.jpg,904,836,103,45,000292.jpg,205,286,false,000293.jpg,ln801
000307.jpg,900,828,103,51,000305.jpg,175,284,false,000306.jpg,ln801
000375.jpg,1026,661,72,30,000371.jpg,126,163,false,000372.jpg,hae003
```

...

Donde **filename** es el nombre del archivo de imagen que contiene a la patente, **x** e **y** son las coordenadas correspondientes al vértice superior izquierdo de la patente respecto a la escena, **w** y **h** corresponden al ancho y al alto de la imagen capturada, **car\_filename** es el nombre del archivo correspondiente a la imagen del vehículo donde se encuentra la patente capturada, **x\_car** e **y\_car** son las coordenadas correspondientes al vértice superior izquierdo de la patente respecto a la imagen del vehículo, **new\_lp** determina si la patente corresponde a la categoría patente 2015 o a la categoría 1994, **scene** corresponde al nombre del archivo de la escena donde se encuentra la patente y por último **characters** corresponde a los caracteres que conforman la patente.

Este dataset, de 75GB en total (gran parte de este número corresponde a los fotogramas denominados escena), es el que se utilizará para entrenar las redes neuronales que se desean estudiar, además el mismo quedará a disposición de la universidad para su posterior usufructo o empleo en materias en los que estos temas sean estudiados, tales como inteligencia artificial. Por otra parte si bien el dataset fue revisado de forma ocular, varias veces y de distintas formas, siempre existirá la posibilidad de errores, duplicados u otros inconvenientes, por lo tanto posiblemente los datos mostrados en este documento podrán variar de forma mínima en el caso de que encontremos algunas imágenes que sean incorrectas o que no se puedan distinguir de una forma fehaciente.

De esta manera y con el análisis mostrado de los datos adquiridos damos por finalizada la primera etapa de esta investigación de forma satisfactoria.

## Redes neuronales

Si bien en el informe de avance se mencionó que se había realizado un curso sobre redes neuronales convolucionales, en este año además se formó parte de la Escuela Latinoamericana de Aprendizaje Profundo (2022) donde adquirimos nuevos conocimientos y limamos algunos otros.

## Entrenamiento de redes neuronales

### A) YOLO

#### *Patentes*

Se utilizó YOLOv3 en su versión más pequeña (tiny), para localizar la patente en una imagen donde se muestra un auto e identificar de qué tipo de patente se trata, es decir si la patente corresponde a la categoría patente 1994 o patente 2015. Es importante aclarar que la red fue entrenada para realizar esta tarea siempre que la imagen de entrada corresponda a un vehículo visto de frente o de atrás. Esto es así debido a que desde un principio planteamos que la detección de una patente tiene que seguir un cierto recorrido o *pipeline* y la red neuronal en este caso estaría solucionando la segunda etapa de este pipeline.



Entonces, para entrenar esta red neuronal, se siguieron los pasos propuestos por el creador de la misma. Así, la arquitectura final de esta red se encuentra dada por darknet-53 pre entrenada con el dataset de Imagenet, a la que le se le agrega la arquitectura tiny-yolov3 al final de la misma.

En este caso, como en los siguientes, hemos utilizado redes pre entrenadas por varios motivos, aunque el principal es que en diversa bibliografía desaconsejan el entrenamiento desde cero de redes tan intrincadas como Yolo por razones de tiempos y complejidad de entrenamiento. Por otra parte las imágenes utilizadas para el entrenamiento, como se mencionó anteriormente, corresponden a vehículos vistos de frente o de atrás, esto lo conseguimos extrayendo del dataset todas las imágenes de patentes que poseen un vehículo asociado. No todas las patentes que se encuentran en el dataset que creamos posee un auto asociado, por lo tanto la cantidad de patentes que utilizamos para entrenar esta red neuronal fue de 24223 contando ambos tipos. De estas patentes tomamos aproximadamente el 30% correspondientes al año 1994 y el 30% correspondientes al año 2015. Estas imágenes fueron utilizadas como testeo, y el 70% de imágenes restantes, de ambas categorías, las utilizamos como imágenes de entrenamiento. Entrenamos la red utilizando darknet, como propone el autor de Yolo, realizando















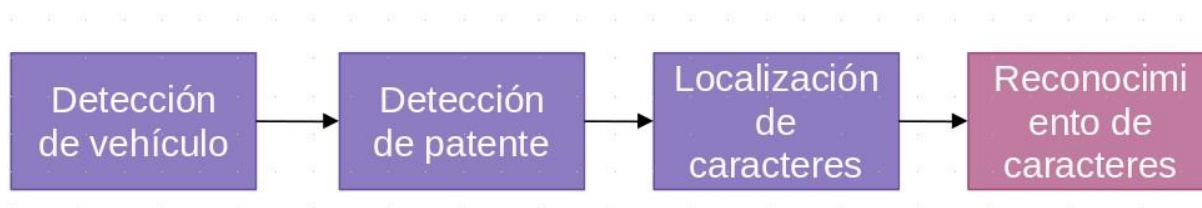








Debido a la decepción que obtuvimos al querer entrenar a Yolo para localizar y reconocer los caracteres *en una sola pasada* decidimos utilizar para la etapa de reconocimientos de caracteres otra red neuronal.

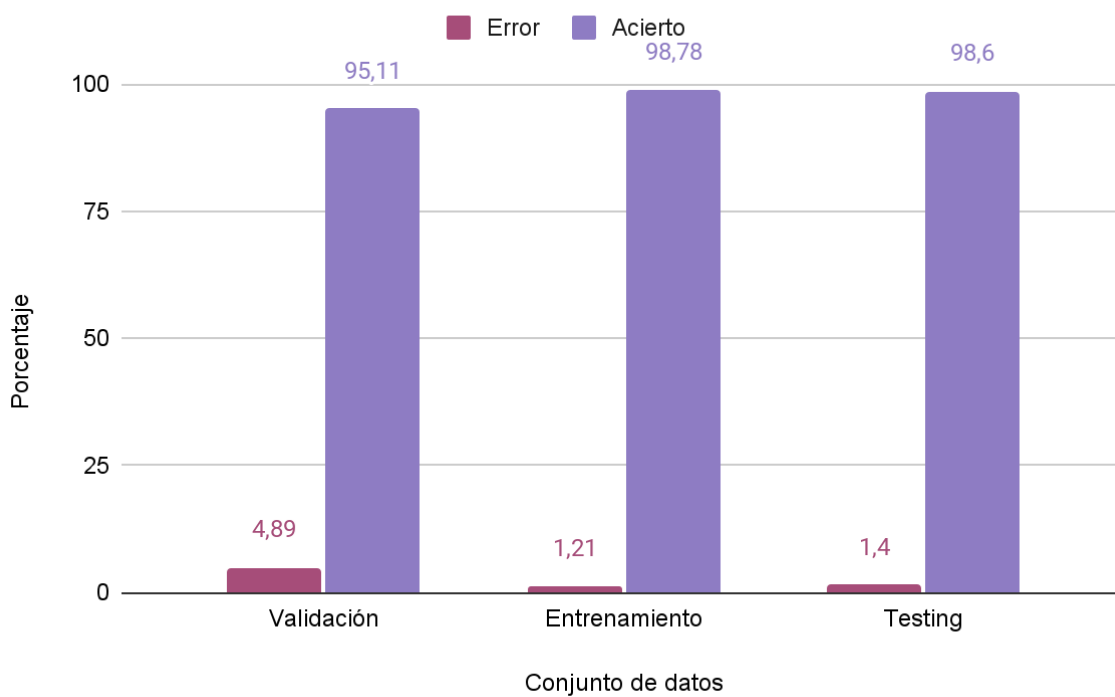


Respecto a la etapa de reconocimiento de caracteres, utilizamos la red neuronal pre entrenada Xception con los pesos de Imagenet. No es el objetivo de esta investigación describir esta red neuronal, aunque explicaremos cuál fue la estructura utilizada y su eficiencia al detectar estas imágenes. Para poder detectar los caracteres y los dígitos de las patentes, utilizamos dos redes neuronales distintas, una que fue entrenada solamente con los dígitos de ambos tipos de patentes, y otra que fue especializada en la detección de los caracteres de ambos tipos de patentes. Esta decisión fue tomada después de observar el fracaso en la localización y reconocimiento de caracteres con YoloV3.

Para el caso de la detección de caracteres, como ya dijimos, fue utilizada una red neuronal Xception con los pesos de Imagenet. Descartamos el segmento de la red neuronal que corresponde a las capas finales (la red neuronal multicapa totalmente conectada que se encuentra al final de esta) y agregamos 3 nuevas capas de redes neuronales totalmente conectadas, con 800 perceptrones por capa. Estas capas junto con las dos últimas capas convolucionales de Xception, fueron las utilizadas para el entrenamiento. El método utilizado fue similar al mencionado anteriormente. Se apartó una cantidad de caracteres cercana al 10% de cada uno de los caracteres de ambos tipos de patentes, este 10% de caracteres son los que utilizamos como validación, el 90% restante se utilizó para el entrenamiento, ejecutando 40 epochs. Por cada epoch utilizamos el 70% de las imágenes para entrenamiento, y luego el 30% restante para el

testeo, por cada epoch el conjunto de imágenes destinadas al entrenamiento y al testeo eran mezclados aleatoriamente, para realizar algo similar a la validación cruzada. Los resultados fueron los que se muestran a continuación

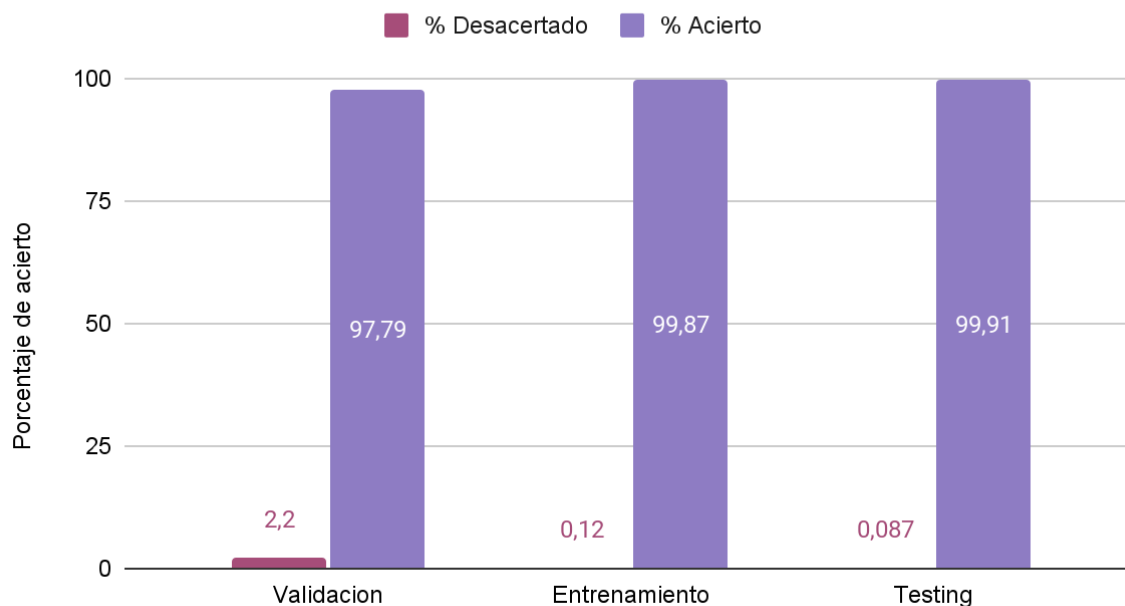
## Eficiencia en la detección de caracteres



La cantidad de imágenes en el conjunto de validación fue de 9747, la cantidad de imágenes en el conjunto de entrenamiento fue de 61615 y la cantidad de imágenes utilizadas en el set de testing fue de 26371. No todas las imágenes de caracteres fueron utilizadas, algunas fueron descartadas para hacer el entrenamiento más fácil y certero. Podríamos concluir que quizás obtener 95,11 % de acierto en el conjunto de validación no es un dato demasiado alentador, más cuando se plantea que la probabilidad de detectar 4 letras consecutivas correctamente es de aproximadamente el 81%. También debemos tener en cuenta la dispersión de las imágenes que se pueden hallar en el dataset, es decir, existen imágenes que distan bastante del ideal de un carácter. Si bien realizamos otra revisión manual de los caracteres antes de comenzar el entrenamiento, descartando varias imágenes, claramente no se removieron todas las que podrían haber producido inconvenientes.

Para el caso de la detección de dígitos, también utilizamos una red neuronal Xception con los pesos de imagenet. En este caso solamente quitamos las últimas capas de la red neuronal, como hicimos en el caso de los caracteres y agregamos 3 nuevas capas de perceptrones cada una de ellas con 800 perceptrones. Sumado a esto incluimos en la etapa de entrenamiento las dos últimas capas convolucionales. El método de entrenamiento fue similar al utilizado en el entrenamiento de caracteres, aunque en este caso la cantidad de epochs fueron aumentadas considerablemente a 120.

## Eficiencia en la detección de dígitos



La cantidad de imágenes en el conjunto de validación fue de 9529, la cantidad de imágenes en el conjunto de entrenamiento fue de 58406 y la cantidad de imágenes en el conjunto de testing fue de 25018. Al igual que en el entrenamiento de los caracteres, no todas las imágenes de dígitos fueron utilizadas, algunas fueron descartadas para hacer el entrenamiento más fácil y certero. Tener un 97,79% de acierto en el conjunto de validación, un 99,87% de acierto en el conjunto de entrenamiento y un 99,91 % de acierto en el conjunto de testing cumple con las expectativas respecto a la etapa de reconocimiento de caracteres.

Adjuntamos los archivos correspondientes a los pesos de estas redes neuronales, junto con un pequeño código en Python que carga las mismas.

En las pruebas hechas con imágenes externas a nuestro set de datos, nos encontramos con que los caracteres no eran reconocidos correctamente en un principio. Después de un largo análisis encontramos que el inconveniente se podía sortear ampliando la imagen de la patente (en nuestro caso 5 veces y utilizando una interpolación cúbica). Algunos de los resultados son los que se muestran a continuación:

Character	X Position	Y Position	Width	Height
A	78	112	95	163
B	172	114	98	155
1	299	112	100	163
2	399	113	99	156
3	496	111	103	160
C	618	112	104	163
D	717	110	102	170



Character	X Position	Y Position	Width	Height
A	40	34	54	95
A	83	32	57	100
7	150	35	56	97
7	205	36	58	98
6	255	35	60	102
D	325	37	58	108
V	383	40	55	105



Character	X Position	Y Position	Width	Height
A	31	111	57	125
A	88	101	57	130
0	161	88	54	124
1	216	80	55	126
9	269	68	54	126
R	339	53	52	122
B	387	48	53	120



Character	X Position	Y Position	Width	Height
A	60	75	58	111
D	114	73	63	111
7	191	73	65	115
6	256	77	61	110
9	312	78	64	112
S	395	79	67	113
I	455	81	63	111



Character	X Position	Y Position	Width	Height
A	39	37	40	63
A	77	37	42	63
0	125	39	39	61
1	162	39	39	61
8	199	42	40	63
E	246	44	41	64
D	284	44	40	65





Character	X Position	Y Position	Width	Height
O	5	24	36	64
B	35	23	36	64
K	68	23	36	64
2	127	24	38	61
7	163	24	31	59
1	194	23	36	62



Character	X Position	Y Position	Width	Height
G	8	25	43	74
T	51	27	35	71
X	87	28	34	72
3	157	29	42	74
4	196	30	35	72
9	228	30	41	74




Character	X Position	Y Position	Width	Height
B	6	35	56	91
K	53	32	55	93
F	104	32	56	94
4	213	29	47	91
5	254	21	55	94
9	305	21	54	93



Character	X Position	Y Position	Width	Height
N	10	23	46	79
M	52	26	44	76
Q	95	26	44	76
3	169	31	44	79
5	210	34	43	78
0	250	35	43	78

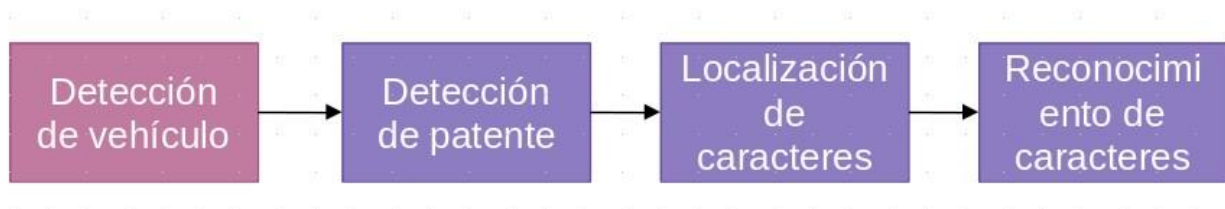


Character	X Position	Y Position	Width	Height
K	12	37	57	94
L	64	37	57	93
W	112	37	58	93
2	204	38	59	90
5	261	39	49	86
2	306	38	59	88



### Automóviles.

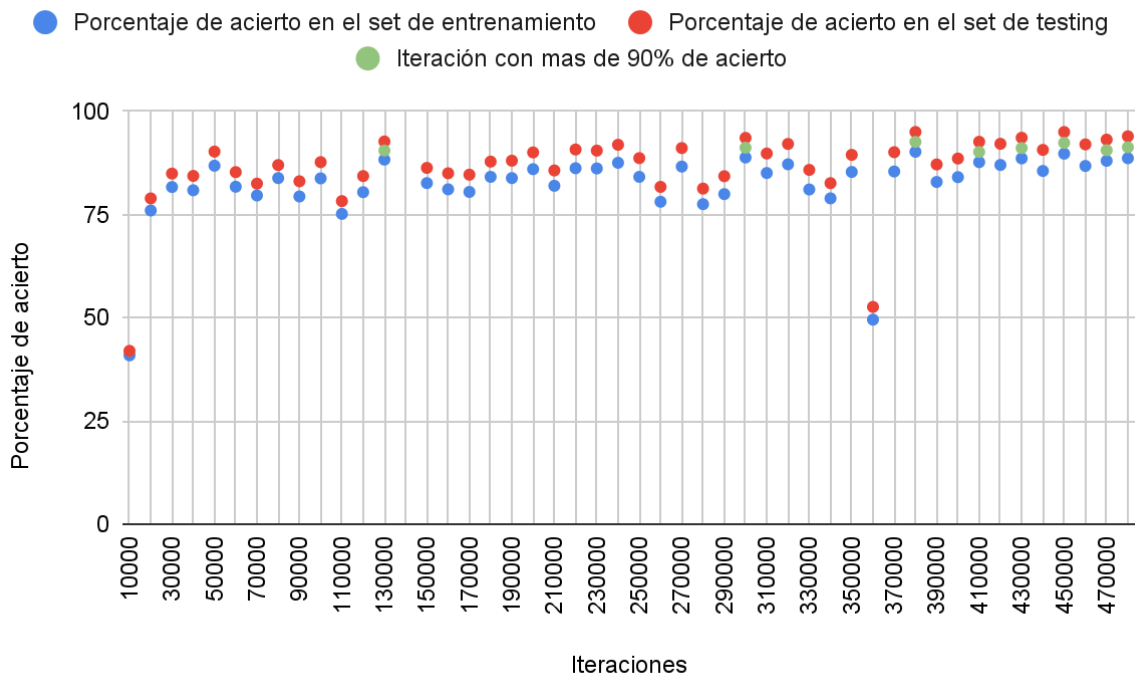
Para el caso de los automóviles procedimos de una forma similar a la metodología utilizada ,y ya descrita anteriormente, de las patentes. Realizado esto completaríamos todas las etapas propuestas del pipelining.



En este caso se dividió todo el set de datos de automóviles en dos sets, uno de entrenamiento ,que contenía el 70% de las imágenes etiquetadas con automóviles, y un segundo de testing que poseía el 30% restante de las imágenes. La única diferencia respecto al entrenamiento de la localización de las patentes, radicó en los anchor boxes utilizados, los cuales no fueron calculados de forma manual sino automáticamente. La cantidad de iteraciones también rondó las 500000.

En un principio, creímos que el reconocimiento de automóviles iba a ser el más sencillo, aunque después de haber adquirido cierta experiencia utilizando YoloV3 en todas las instancias anteriores y comprendiendo ciertas mecánicas de su funcionamiento notamos que habíamos cometido un error realmente grave al etiquetar los automóviles. El inconveniente consistió en que Yolo intenta discernir qué partes de la imagen corresponden a la clase a buscar, pero también intenta hacer lo propio con las partes de la imagen que no corresponde a ninguna clase, es decir esta red neuronal *aprende* de toda la escena y no únicamente con las clases con las que se pretende entrenar. Teniendo esto en cuenta, en el dataset que construimos, no siempre

etiquetamos todos los automóviles que se podían observar en la escena, sino que existieron automóviles que no consideramos que eran aptos para nuestro dataset, y por lo tanto sencillamente fueron descartados. Es decir, aquellos automóviles que se vieran de frente o desde atrás, cercanos a la cámara y que su patente fuera legible, eran preferidos sobre los vehículos que se encontraban lejos en la escena, de costado, o que su patente no pudiera ser distinguida correctamente. Este sesgo puede observarse en el gráfico de dispersión de vehículos presentado anteriormente. Con esto en consideración, comenzamos el entrenamiento de otra red neuronal Yolo y los resultados que obtuvimos fueron los que se muestran a continuación:

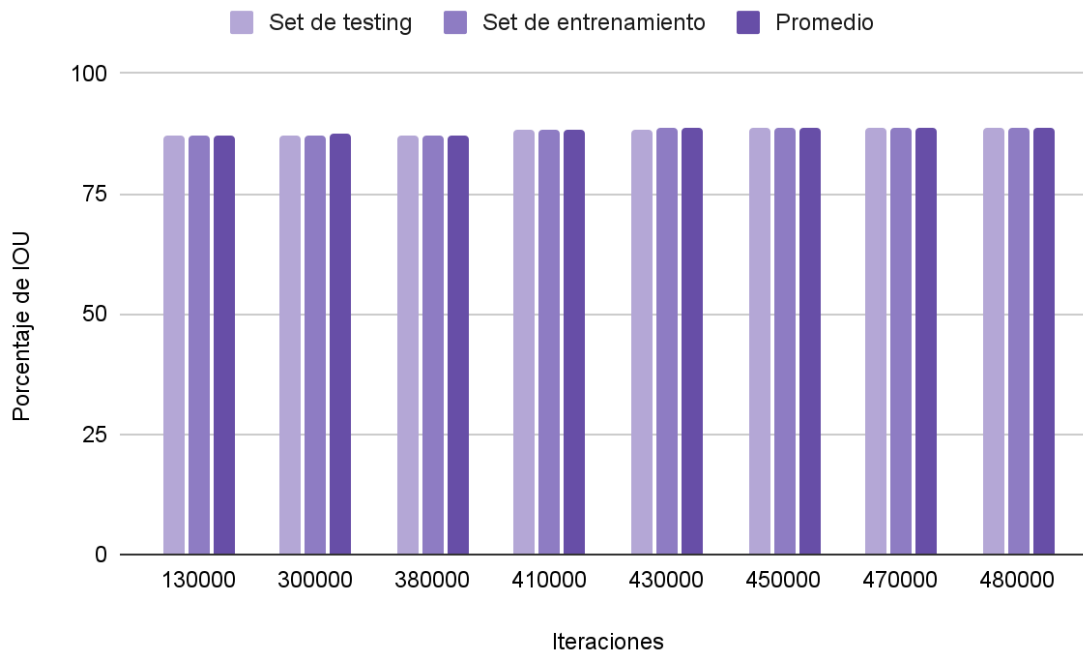


Teniendo en cuenta lo que mencionamos anteriormente, para nuestro asombro la red neuronal tuvo una convergencia aceptable, sin demasiados sobresaltos. Este comportamiento se la atribuimos a dos razones:

1. La primera razón y no menos importante, es que estamos utilizando una red neuronal pre-entrenada, como ya hemos mencionado, con el dataset de Imagenet. El entrenamiento de una red neuronal con este dataset, hace que la red neuronal sea pasible de detectar automóviles, aunque no exactamente los que deseamos en nuestro caso.
2. Los vehículos con los que entrenamos esta red neuronal tienen ciertas características en común (como ya hemos explicado tienen un sesgo). Estas características son las que posiblemente sean acentuadas en el entrenamiento y por lo tanto explican en alguna medida el buen desempeño mostrado en el gráfico anterior.

Teniendo en cuenta estas posibles causas, procedimos a elegir la iteración con mejor desempeño según nuestro criterio. De todas las iteraciones seleccionamos únicamente las que

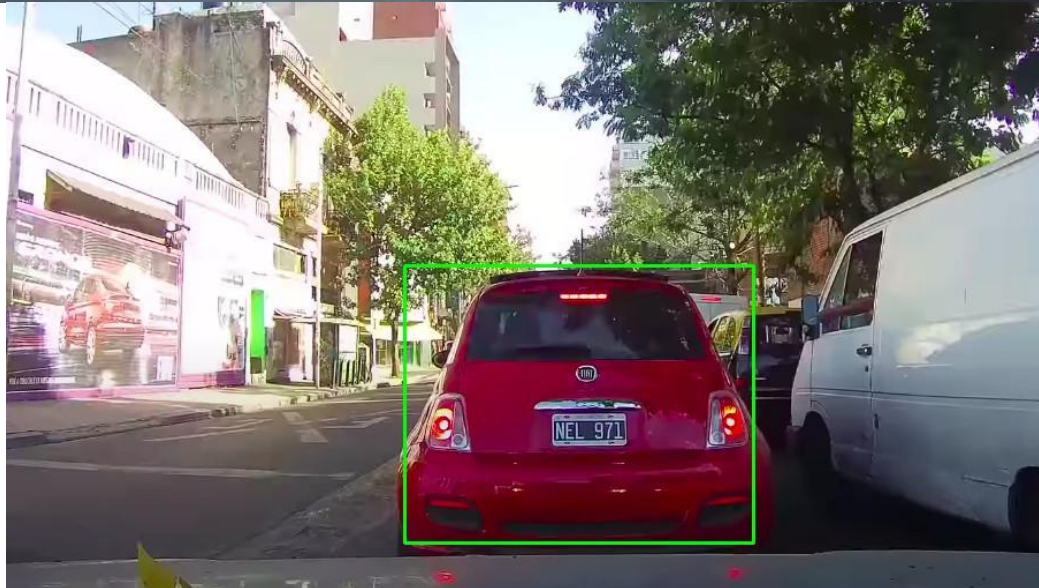
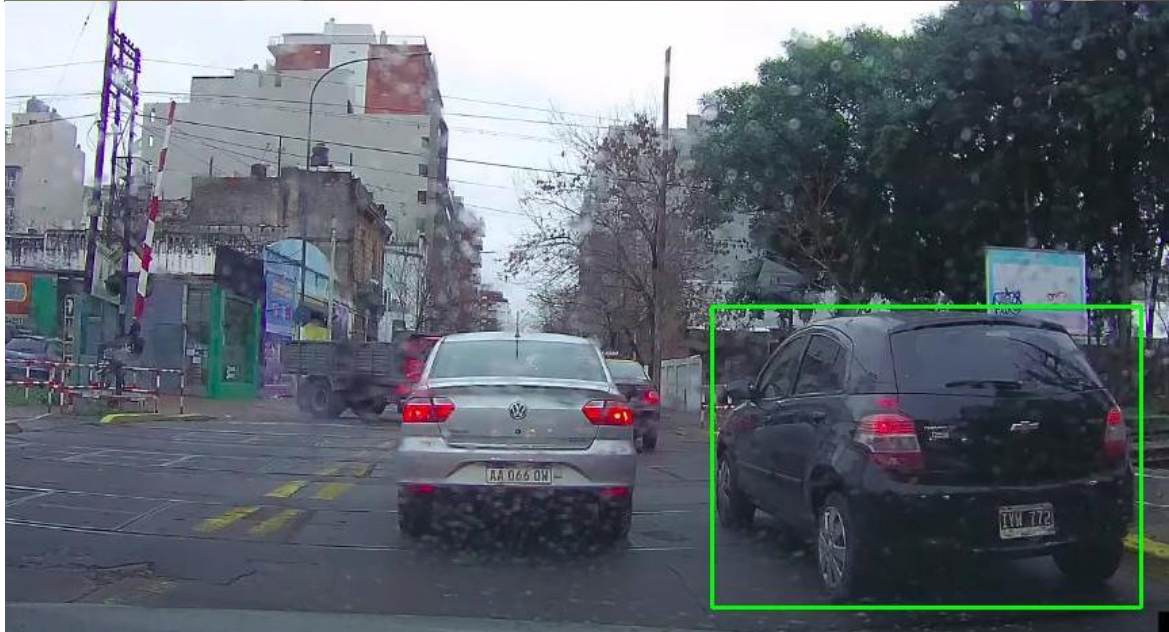
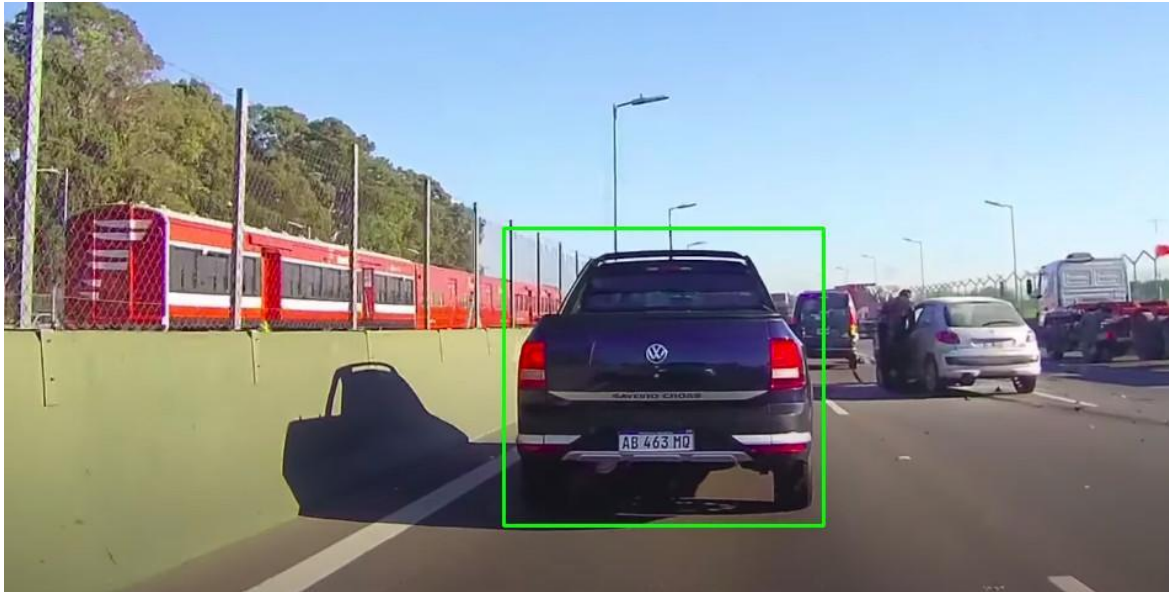
poseían un porcentaje de acierto mayor o igual al 90% (mostrados en verde en el gráfico). De estas iteraciones observamos cual poseía mejor IOU.



Al graficar los IOU podemos percatarnos que las diferencias en los promedios de IOU no son realmente notables. Sin embargo la iteración que posee mejor IOU es la 480000, aunque esta frente a la iteración 380000 posee porcentaje de acierto ligeramente un menor, por lo tanto siendo que las diferencias entre IOU no son muy notorias consideramos que posee mejor desempeño la iteración 380000.

Adjuntamos los archivos correspondientes a la configuración de la red neuronal Yolo necesarios para poder utilizarla, junto con el archivo de los pesos correspondientes a la iteración 380000, iteración que consideramos que obtuvo mejor performance. Algunos resultados de cómo se desempeña esta red neuronal con los pesos obtenidos, sobre imágenes no utilizadas ni en el entrenamiento ni en el testeo se pueden observar a continuación:

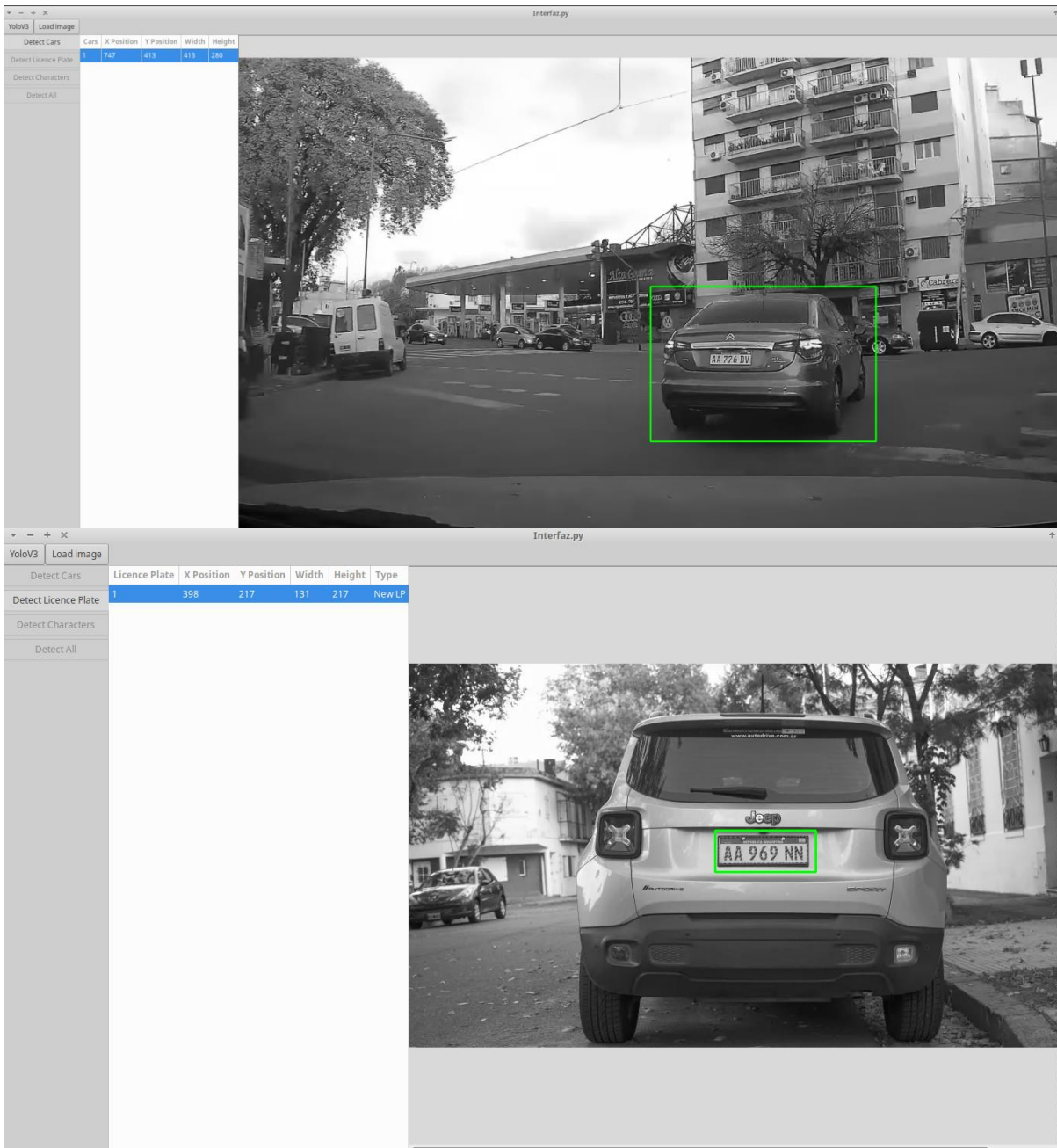


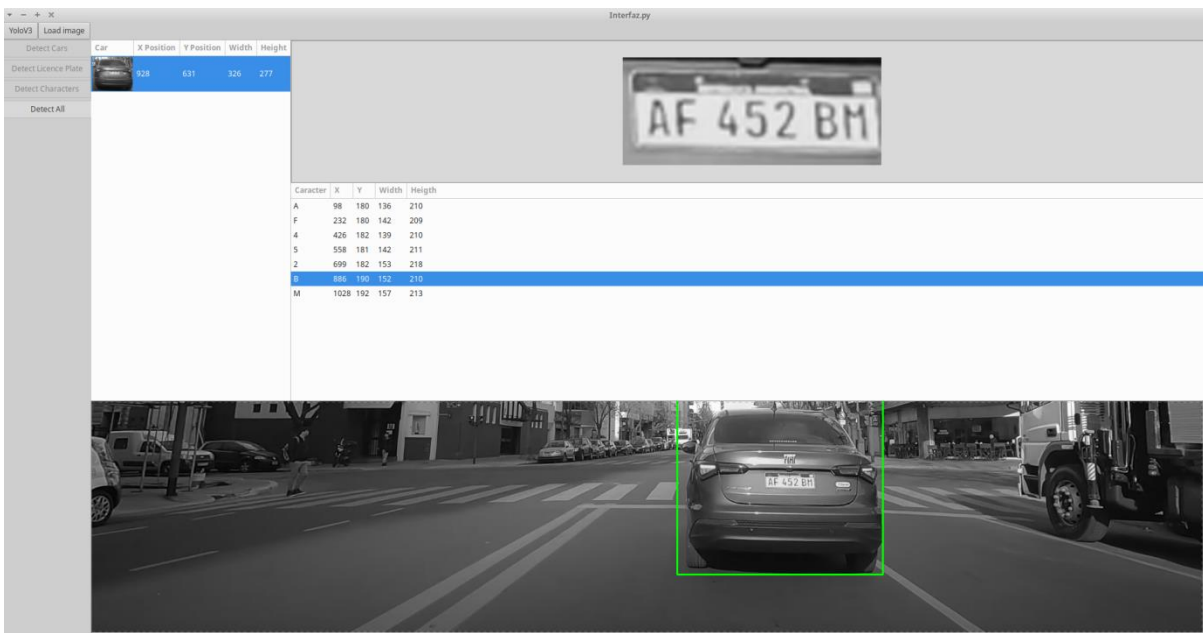


Para poder llegar a estos resultados, en el programa de pruebas por alguna razón, la tolerancia en la detección tuvo que ser disminuida considerablemente.

## Herramienta de prueba

Para realizar las pruebas con las que capturamos los resultados anteriormente mostrados, se confeccionó un programa de pruebas interno que permitió la búsqueda de autos en una escena, localizar una patente en la imagen de un automóvil, detectar los caracteres de una patente y por último cumplir con el pipeline completo.





## Implementación en un sistema embebido

### *Análisis y adaptación de una red neuronal en un sistema embebido*

Quando Alan Turing comenzó a imaginar su famosa máquina computadora universal, en teoría el disparador de esta idea fue una charla que fue impartida por el matemático Max Newman en el año 1935, en la cual dicho matemático dijo que sería posible crear una máquina que realizara procedimientos sistemáticos. Newman estaba haciendo referencia a los procedimientos repetitivos que en algunos casos abundan en matemáticas para resolver ciertos problemas, como el algoritmo de la suma, la resta, la multiplicación, división, raíces y tantos otros. Por esas fechas, la palabra computador estaba atribuida a las personas que solamente realizaban cálculos en diversos ámbitos, como el de la investigación, negocios y gobierno, y para realizar este trabajo, si



bien se necesitaba estar bien instruido, no se requería de ninguna forma **inventiva, intuición, creatividad ni entendimiento**.<sup>5</sup>

Esta puesta en común es necesaria e interesante para poner en consideración una realidad, que es que la arquitectura de computadoras (teniendo en cuenta la arquitectura de Von Neumann y de Harvard) que comúnmente utilizamos, no fueron concebidas para el fin de tener ni **inventiva, intuición, creatividad ni entendimiento** y esto, que parece una verdad obvia, tiene sus repercusiones.

Al intentar programar para una computadora, para las arquitecturas ya mencionadas, un programa que de alguna forma pueda simular inventiva, creatividad, intuición o entendimiento, sucumbiremos ante el problema de la eficiencia, por el solo hecho de que estas arquitecturas, y el concepto con el cual fue creada la máquina computadora universal de Turing no contemplan la solución a estos problemas.

Incluso adaptar una red neuronal, si bien de alguna forma trata de paliar esta verdad, deviene en un recorte de las capacidades de la misma en desmedro de la precisión o el buen desempeño de la red.

Por lo tanto es posible que una red neuronal pueda funcionar en un sistema embebido, aunque el desempeño que se vaya a obtener no sea aplicable a tareas en las cuales se necesite responder en tiempo real. Esto siempre y cuando el sistema embebido no ofrezca otro tipos de arquitecturas a utilizar como GPUs o TPUs. La primera de estas unidades no tuvo por objetivo realizar tareas que posean inventiva, intuición o creatividad, aunque comienza a tener cierta semejanza a como esta compuesto un cerebro humano. En este poseemos millones de neuronas que disponen de un grado de procesamiento muy pequeño pero que en conjunto gozan de un gran poder permitiendo realizar tareas que tienen cierto grado de complejidad como las que venimos mencionando, y las GPUs poseen cientos de unidades funcionales, que funcionan de forma paralela y resuelven tareas en conjunto. Las TPUs por otro lado, tienen también cierta similitud con un cerebro cuando hablamos de las unidades de procesamiento, en este caso también son numerosas y con bajo poder de cálculo respecto a una CPU. Así como las TPUs fueron desarrolladas para solucionar un problema de eficiencia en lo que respecta al entrenamiento de las redes neuronales, otras arquitecturas también persiguen este fin pero utilizando una tecnología no digital, sino analógica como Mythic.<sup>6</sup>

Teniendo esto en cuenta, existen sistemas embebidos que poseen algunas de estas unidades especiales, que permiten acelerar la etapa de reconocimiento, aunque no son recomendables para el entrenamiento. Para estos sistemas embebidos o sistemas especiales, de bajas prestaciones se han adaptado redes neuronales, perdiendo en una proporción sus capacidades, pero siendo utilizables como Mobilenet o Yolo-tiny, la cual hemos entrenado en la presente investigación. Y en el caso de que aun estos sistemas embebidos no logren ejecutar en tiempo y forma una determinada red neuronal con el hardware disponible, pueden adaptarse con hardware externo, como el "Neural ComputerStick" desarrollado por Intel, el cual es descrito como unidades de procesamiento de visión o VPU.<sup>7</sup>

En nuestro caso intentamos utilizamos una placa Raspberry Pi 3 para ejecutar el pipeline propuesto, aunque rápidamente pudimos constatar (por todo lo descrito anteriormente) que es imposible ejecutar, al menos, las redes YoloV3 entrenadas en esta investigación, en un tiempo

---

<sup>5</sup> Alan Turing-El pionero de la era de la información-B. Jack Copeland.

<sup>6</sup> <https://mythic.ai/>

<sup>7</sup> <https://www.intel.com/content/www/us/en/products/sku/140109/intel-neural-compute-stick-2/specifications.html>

razonable, sin la utilización de un hardware externo dedicado. Sin embargo fue posible, de forma exitosa, la ejecución de todas las redes neuronales YoloV3 en un sistema embebido, en la placa de desarrollo NvidiaJetson TK1.

Para lograr esto desarrollamos un pequeño sistema cliente-servidor, donde el sistema embebido trabaja como servidor, esperando las solicitudes de un cliente que se ejecuta en una computadora convencional. El cliente ejecuta los pasos del pipeline propuesto (menos el reconocimiento de dígitos y caracteres que se ejecuta de forma local debido a las dimensiones de Xception y a la imposibilidad de poder implementarlo en la placa mencionada), y al necesitar localizar una patente en un vehículo, un vehículo en una escena, o localizar los caracteres en una patente envía una petición al sistema embebido, esperando el resultado de la localización y mostrándolo en pantalla (para el cliente utilizamos la herramienta de prueba descrita en párrafos anteriores).

De esta forma podemos concluir que es posible implementar YOLOv3 en un sistema embebido, como el NvidiaJetson TK1 y por lo tanto poder ejecutar en gran medida el pipeline propuesto. Respecto a Xception, y a la detección de caracteres, creemos que la red neuronal está sobredimensionada para la tarea que debe realizar y que con una red neuronal más pequeña sería posible ejecutar todo el pipeline en el sistema embebido sin mayores inconvenientes.

## Conclusiones

En primer lugar, pudimos concretar correctamente el dataset propuesto, aunque nos consumió mucho más tiempo del planeado. Esto se debió a diferentes problemas que surgieron en la creación del mismo, sumando a las demoras de la extracción de las distintas imágenes que conforman el dataset y a su posterior depuración. Anteriormente pensábamos que 5000 patentes iban a ser suficientes para empezar el entrenamiento, pero al observar el tiempo que nos consumía la creación del dataset, nos enfocamos en terminar esta tarea sobre otras. Más allá de estos inconvenientes, los integrantes de este grupo de investigación nos encontramos satisfechos con los resultados del mismo.

Por otra parte entrenamos la red neuronal YoloV3 como explicamos anteriormente. Si bien conocíamos esta red neuronal, nos encontramos con diversos inconvenientes cuyas resoluciones fueron explicadas en este documento, consumiendo esfuerzo y tiempo de entrenamiento. Nos hubiera gustado, al menos, poder entrenar Mobilenet para poder así realizar comparaciones entre las dos redes, pero debido a los tiempos que nos ocupó la creación del dataset, los entrenamientos y los tests nos fue imposible. Más allá de esto los resultados en general fueron satisfactorios en el desempeño de YoloV3, y las detecciones realizadas con elementos que no corresponden al dataset nos parecen aceptables.

Respecto al análisis de si es posible implementar una red neuronal capaz de ejecutar las tareas propuestas en un sistema embebido, llegamos a la conclusión de que es posible, siempre y cuando exista hardware dedicado, sino es prácticamente imposible. Existen redes neuronales que están pensadas para ejecutar en hardware con pocas prestaciones como YoloV3 o Mobilenet, pero incluso así es necesario tener hardware dedicado. Pudimos implementar, no todo el pipeline propuesto, pero si una importante parte del mismo en un sistema embebido NvidiaJetson TK1, placa que poseíamos con anterioridad, en donde la misma funciona como un servidor, atendiendo

peticiones de un cliente para hacer todas las predicciones que involucren el uso de YoloV3. Poder correr Xception nos fue imposible en dicha placa hasta el momento.

Debido a las demoras mencionadas, y a que pudimos enfocarnos en una red neuronal, no fue posible llevar a cabo el documento detallado explicando las diversas redes neuronales que íbamos a entrenar.

Por último, y no menos importante, resultado de esta investigación fue presentado en el Workshop de Investigadores en Ciencias de la Computación 2023, el trabajo “Confección de dataset de patentes de automóviles argentinos para entrenamiento de redes neuronales”, en abril de 2023.