

Índice

[Sobre el grupo 3](#_Toc135822509)

[Integrantes: 3](#_Toc135822510)

[Organización del grupo: 3](#_Toc135822511)

[Planificación: 3](#_Toc135822512)

[Subtareas: 3](#_Toc135822513)

[Sobre el proyecto 3](#_Toc135822514)

[Introducción 3](#_Toc135822515)

[Herramientas 4](#_Toc135822516)

[Lenguaje PYTHON 4](#_Toc135822517)

[Paquete PYTORCH y algoritmo YOLO 4](#_Toc135822518)

[IDE PYCHARM 4](#_Toc135822519)

[¿Qué son las redes convolucionales? 4](#_Toc135822520)

[Breve historia de las redes convolucionales 5](#_Toc135822521)

[Importancia en la actualidad 5](#_Toc135822522)

# Sobre el grupo

## Integrantes:

* Araya, Matías
* Osorio, Pablo
* Sandoval, Agustín

## Organización del grupo:

Al ser un proyecto en etapa de gestación, y al no tener experiencia en este tipo de trabajos, iremos viendo qué roles cumplirá cada integrante según sus capacidades.

A medida que avance el proyecto aparecerán obstáculos y objetivos a corto plazo por cumplir. Mientras tanto, iremos dividiendo la idea principal en problemas más pequeños. Esto nos permitirá planificar el avance del proyecto y dividir las subtareas dependiendo del rol que ocupe cada integrante.

Hoy por hoy, nuestro objetivo a menor plazo es presentar de la mejor forma posible. A continuación, lo que tenemos hecho.

### Planificación:

* Definimos el tipo de objetos a detectar y los requerimientos necesarios.
* Seleccionamos los tipos de herramientas a utilizar (biblioteca, entorno de desarrollo, imágenes a utilizar).
* Recopilamos datos.
* Definimos la arquitectura del modelo.

### Subtareas:

* Adquirir un conjunto de datos adecuado para entrenar el modelo (Recopilación de imágenes).
* Preparación de datos para alimentar el modelo.
* Entrenamiento del modelo.
* Pruebas del modelo (probar su funcionamiento en imágenes que no haya visto).
* Integrar el modelo en la aplicación a utilizar.

**(Actualización 24/5)**

Luego de varias reuniones, logramos avanzar y entender las dimensiones más realistas a las que deseamos aspirar para este proyecto. Ya definimos el logo y el nombre “comercial”, el lenguaje y las librerías, también se han ido definiendo más o menos los roles del grupo (Matías se está encargando de GitHub, Pablo de Jira y Agustín del informe y las presentaciones) y hemos avanzado bastante con el código, en el cual trabajamos todos por igual.

Resta decidir y consultar con el profesor la forma de presentar el software una vez terminado. Los avances se ven reflejados desde la presentación hasta el código.

# Sobre el proyecto

## Introducción

En los últimos años, la detección de objetos se ha convertido en un área de investigación y desarrollo crucial en el campo de la visión por computadora. La capacidad de identificar y localizar objetos en imágenes o vídeos es fundamental para una amplia gama de áreas, como la conducción autónoma, la vigilancia, sector de salud, el reconocimiento facial, la realidad aumentada, entre otras.

El propósito de este proyecto es implementar un sistema de detección de objetos utilizando el lenguaje de programación Python ya que se ha convertido en uno de los lenguajes de programación más populares en el ámbito de la inteligencia artificial y el aprendizaje automático debido a su simplicidad, versatilidad y una amplia gama de bibliotecas especializadas disponibles, y la arquitectura YOLO (You Only Look Once) junto con PyTorch que es una poderosa biblioteca de aprendizaje automático que proporciona herramientas para construir y entrenar redes neuronales de manera eficiente.

Como dijimos previamente, utilizaremos Python y PyTorch (entre otras librerías, pero en especial PyTorch) para implementar el algoritmo YOLO, que es conocido por lograr un equilibrio entre precisión y velocidad, lo que lo convierte en una opción ideal para aplicaciones en tiempo real. Además, aprovecharemos las numerosas librerías disponibles en Python para facilitar el procesamiento de imágenes y la visualización de resultados.

El objetivo final de este proyecto es desarrollar un sistema de detección de objetos preciso y eficiente que pueda utilizarse en diversas situaciones o áreas de la vida.

## Herramientas

### Lenguaje PYTHON

****: Para poder desarrollar este módulo es necesario usar herramientas de libre disponibilidad. Por una parte está el lenguaje de programación que mejor se adapta a los objetivos, que es Python. Python es un lenguaje de programación interpretado cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Es un lenguaje de programación multi-paradigma, ya que soporta orientación a objetos, programación imperativa y programación funcional. Es administrado por la “Python Software Foundation” y posee una licencia de código abierto, denominada Python Software Foundation License, compatible con la Licencia pública general de GNU.

### Paquete PYTORCH y algoritmo YOLO

****: Es una biblioteca de aprendizaje automático​ de código abierto basada en la biblioteca de Torch, utilizado para aplicaciones como visión artificial y procesamiento de lenguajes naturales, principalmente desarrollado por el Laboratorio de Investigación de Inteligencia Artificial de Facebook (FAIR). Es un software libre y de código abierto liberado bajo la Licencia Modificada de BSD.

**** : El algoritmo You Only Look Once (YOLO), es un sistema de código abierto del estado del arte para detección de objetos en tiempo real, el cual hace uso de una única red neuronal convolucional para detectar objetos en imágenes o videos.

### IDE PYCHARM

****: PyCharm es un entorno de desarrollo integrado (IDE) que se utiliza para la programación en Python. “PyCharm Community Edition” se distribuye bajo licencia Apache. La licencia Apache es una licencia de software libre escrito por “Apache Software Foundation”, (ASF).

## ¿Qué son las redes convolucionales?

Las redes neuronales convolucionales (convolutional neural networks, CNN o ConvNets) son una herramienta reciente de la tecnología. Son un tipo de red neuronal, que surge como variación del perceptrón multicapa, la diferencia se encuentra en el uso de capas de convolución y de subsampling, ideas que desarrollaremos más adelante. Están especialmente diseñadas para el procesamiento de datos de dos dimensiones, sobre todo reconocimiento de imágenes y señales de voz en espectrogramas, también pueden ser modificadas para la realización de tareas en una dimensión o varias dimensiones.

De manera que suponemos que tienes Facebook o lo conoces, así que te habrás fijado en que cuando vas a etiquetar a alguien te sugiere personas y, normalmente, acierta. Así te familiarizarás con el concepto **convnets**.

Asimismo, si tienes un iPhone, habrás visto la carpeta **Personas** dentro de la galería. Te preguntarás cómo hacen eso, cómo son capaces de encontrar la cara en una imagen y, no contentos con eso, también son capaces de agrupar todas las caras que pertenecen a una determinada persona. Una vez más: **convnets**.

Por otra parte, quizás hayas oído hablar de los coches autónomos que son capaces de **leer** las señales de tráfico, incluso detectar si hay una persona cruzando la calle. Otra vez: **convnets**.

## Breve historia de las redes convolucionales

Las redes neuronales convolucionales comienzan en la década de 1980 con los trabajos pioneros de Kunihiko Fukushima, quien propuso una arquitectura llamada "Neocognitron". El Neocognitron estaba inspirado en el sistema visual del cerebro humano y utilizaba capas de neuronas convolucionales y de submuestreo para extraer características jerárquicas de las imágenes.

Sin embargo, fue en la década de 1990 cuando Yann LeCun, junto con otros investigadores, introdujo una variante más eficiente y práctica de las redes neuronales convolucionales conocida como LeNet. LeNet fue diseñada específicamente para el reconocimiento de dígitos escritos a mano y se convirtió en un hito en el campo del reconocimiento de patrones.

A medida que avanzaba la década de 2000, el aumento en la disponibilidad de datos y la capacidad computacional permitieron el desarrollo de arquitecturas más profundas y poderosas. En 2012, Alex Krizhevsky y su equipo presentaron la red neuronal convolucional AlexNet, que ganó el desafío de clasificación de imágenes ImageNet con un rendimiento sorprendente. AlexNet demostró la capacidad de las redes neuronales convolucionales para superar a los métodos tradicionales en tareas de visión por computadora.

Desde entonces, las redes neuronales convolucionales han experimentado un rápido desarrollo y se han aplicado con éxito en diversas áreas, como el reconocimiento de objetos, la segmentación de imágenes, el procesamiento de lenguaje natural y la conducción autónoma.

## Importancia en la actualidad

Existe una enorme cantidad de situaciones donde se involucra la detección de objetos. Se puede pensar, por ejemplo, en un perímetro amplio alrededor de una gran instalación. En este caso simples sensores de movimiento no son posibles porque darían falsas alarmas, mientras un sistema que detectara personas no fallaría en el caso de que una persona intente cruzar dicho perímetro. O quizás la demostración más común y a la mano es el reconocimiento facial que viene integrado en la mayoría de los smartphones contemporáneos, cada vez que tomamos una foto de una persona, bien sea un selfie o un retrato normal, nuestra cámara utiliza un algoritmo de detección facial para asegurarse de que las caras en la imagen no se distorsionen ni se desenfoquen. Variaciones de este mismo algoritmo es lo que soporta el etiquetado en plataformas sociales como Facebook e Instagram, etc. Pero también se utiliza mucho como sistema de seguridad de bloqueo y así proteger nuestra información.

Hoy en día el reconocimiento de objetos es el núcleo de la mayoría de los programas y software de inteligencia artificial basados en la visión, ya que juega un papel muy importante en la comprensión de la escena, que es popular en casos como dijimos anteriormente de seguridad y redes, pero también como casos muy importantes como: transporte (seguridad vial), medicina, entre otros.