

# Conteo de vehículos usando YOLO - You Only Look Once

FACULTAD DE INGENIERÍA  
UNIVERSIDAD DE ANTIOQUIA  
Departamento de Ingeniería de Sistemas

**Docente:**

David Stephen Fernandez Mc Cann  
[david.fernandez@udea.edu.co](mailto:david.fernandez@udea.edu.co)

Semestre 2021-1

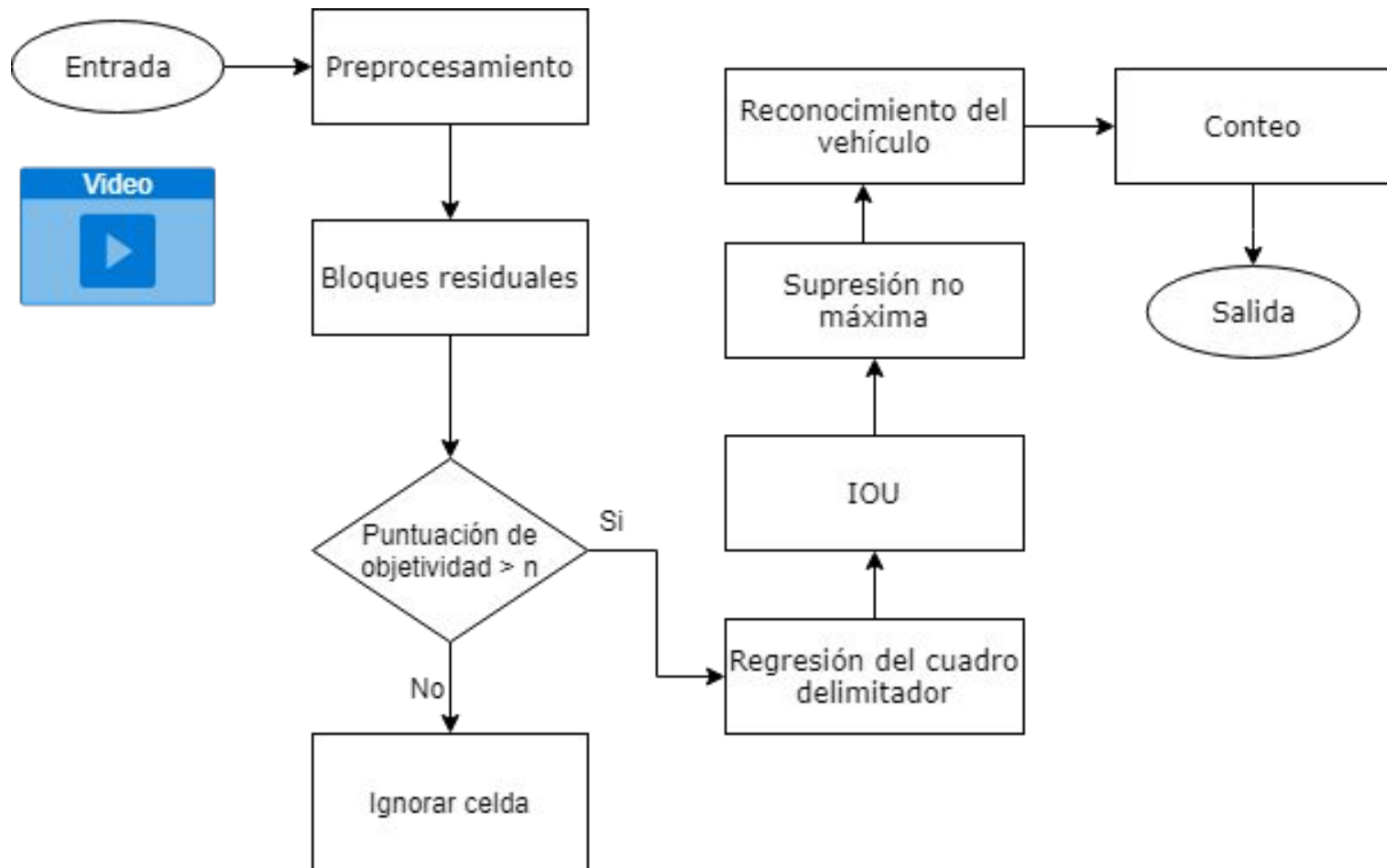
Alejandro Gallego Durango  
[wildey.gallego@udea.edu.co](mailto:wildey.gallego@udea.edu.co)

Lina María Uribe  
[lina.uribem@udea.edu.co](mailto:lina.uribem@udea.edu.co)

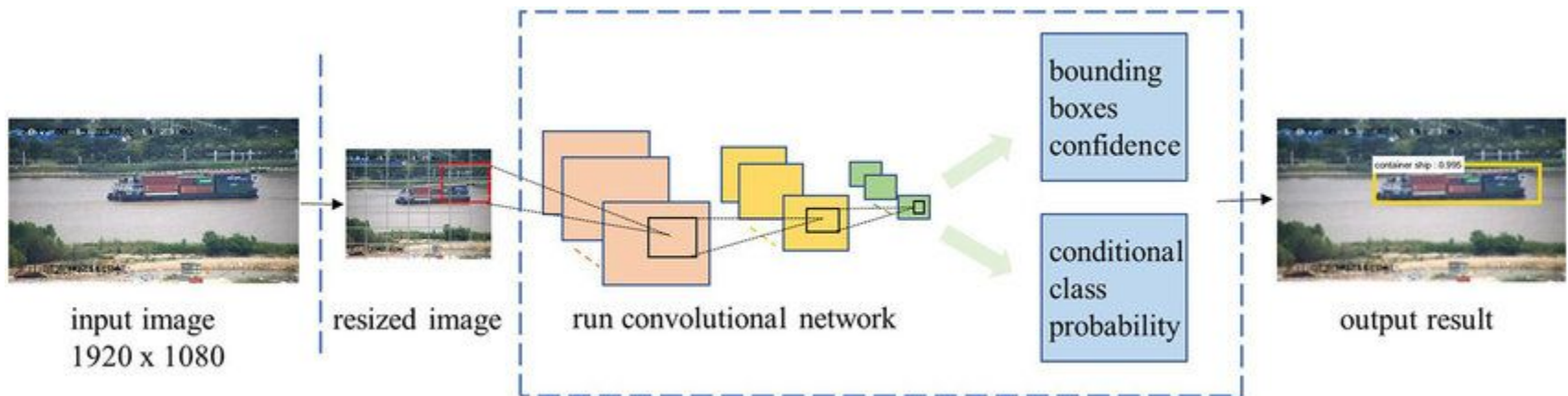
# Descripción del Problema

- **Problema humano:** Presentar un video en el que se realice un conteo de los vehículos que allí aparecen de forma automática y novedosa.
  - **Problema técnico:** Detección de objetos usando el algoritmo YOLO para realizar el conteo de vehículos.
  - **Trabajos relacionados:**
    - The PASCAL Visual Object Classes (VOC) Challenge [1]
    - Deep Residual Learning for Image Recognition [2]
    - Feature Pyramid Networks for Object Detection [3]
    - Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [4]
-

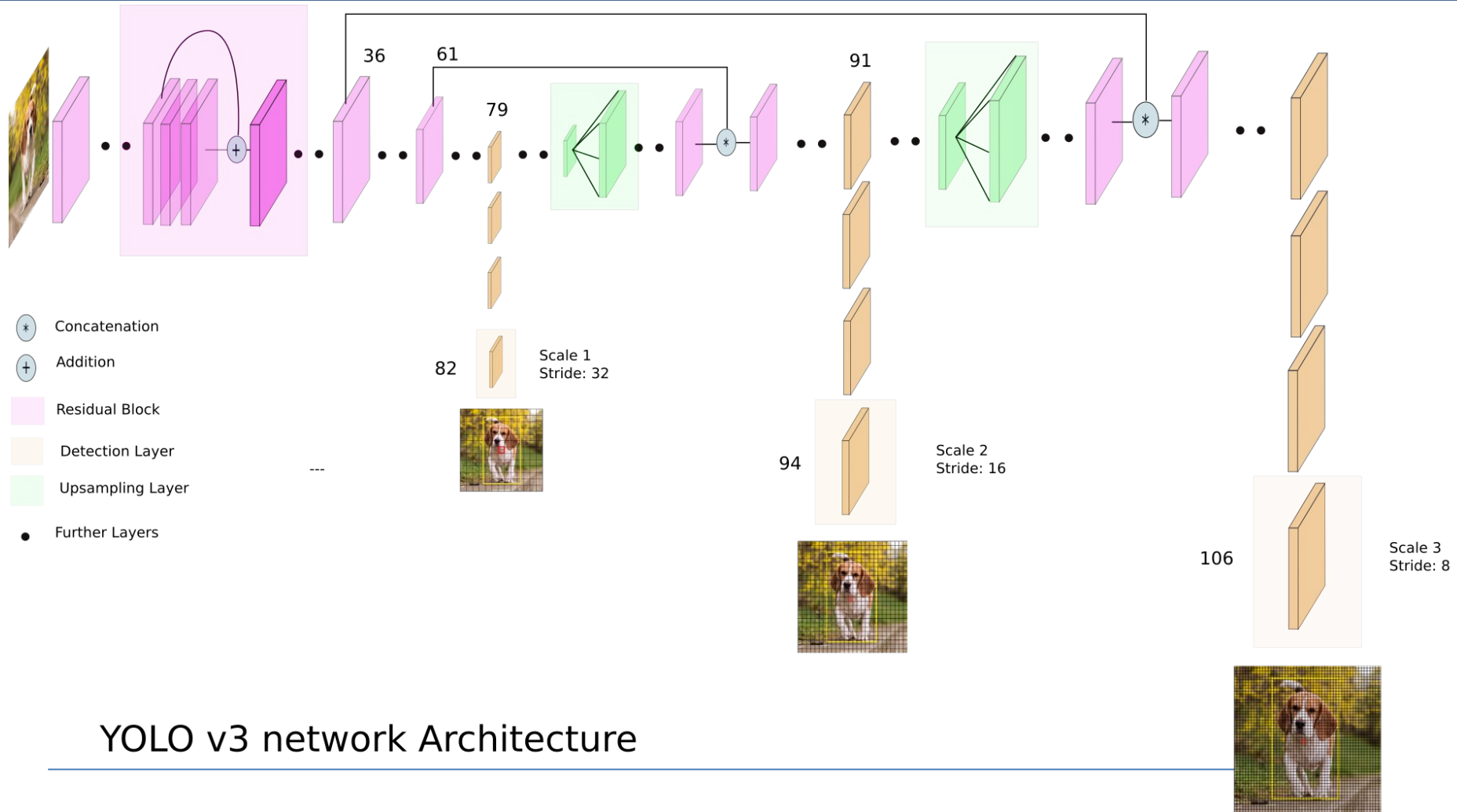
# Propuesta de Solución: Ruta a seguir



# Propuesta de Solución: Conceptos

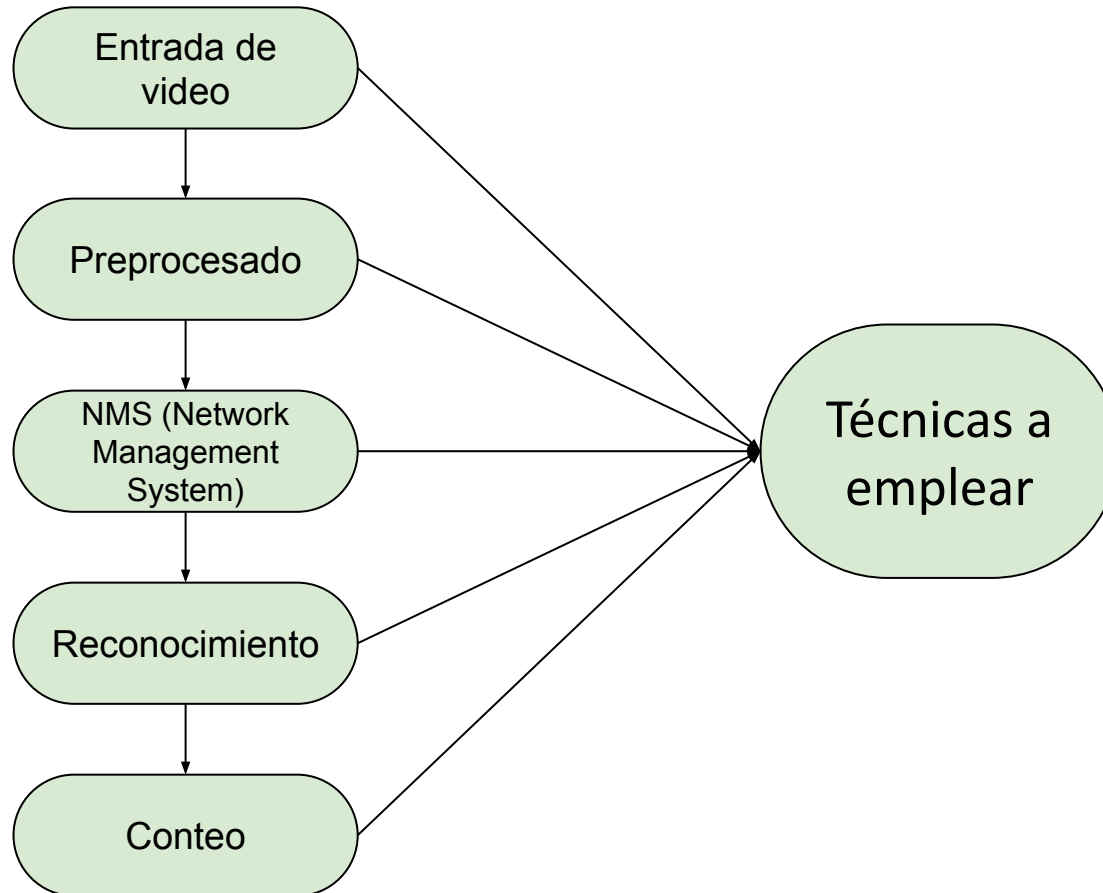


# Propuesta de Solución: Arquitectura YOLO



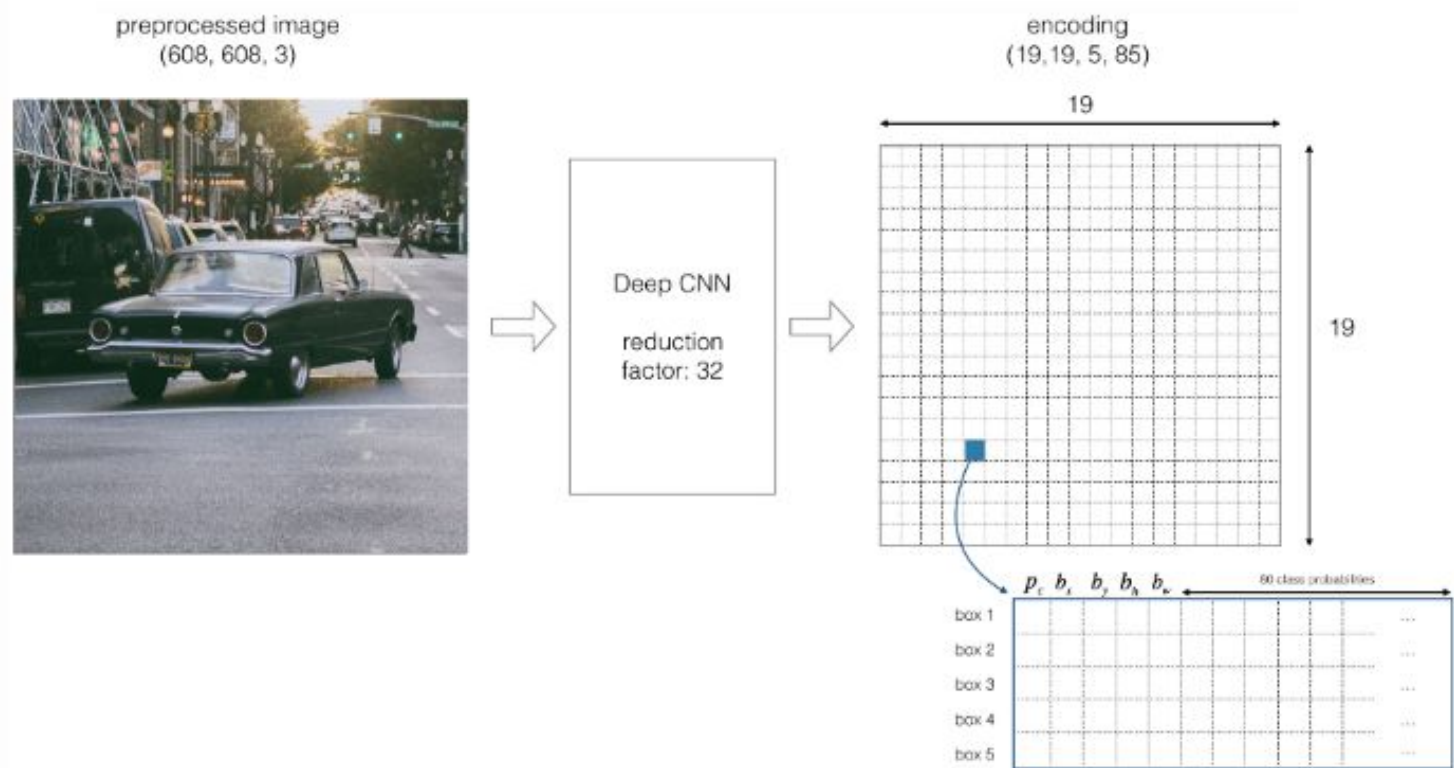
YOLO v3 network Architecture

# Propuesta de Solución: Conceptos



# Técnicas a emplear

- Técnica 1: Bloques residuales



# Técnicas a emplear

- Técnica 1: Bloques residuales



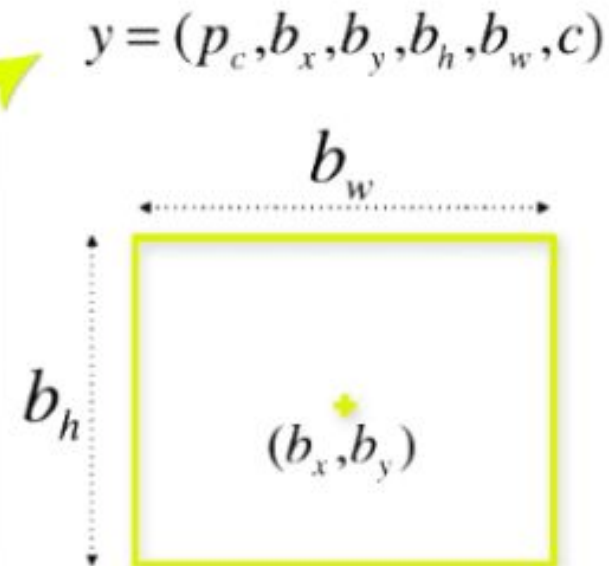
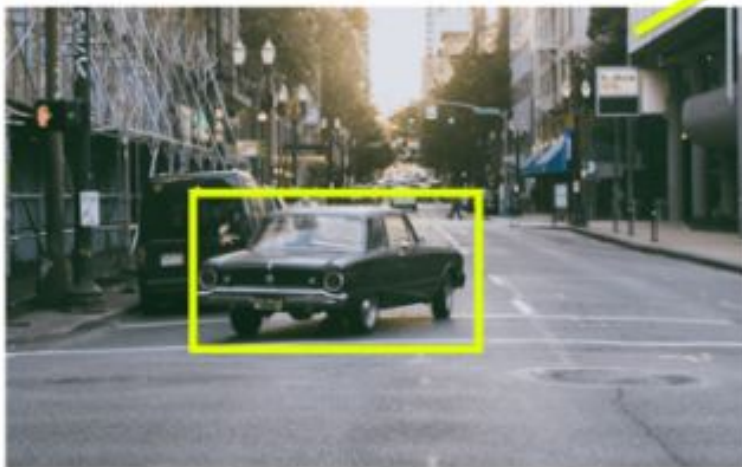
car  
road sign  
tree  
traffic light  
sky  
background

$$score_{c,i} = p_c \times c_i$$



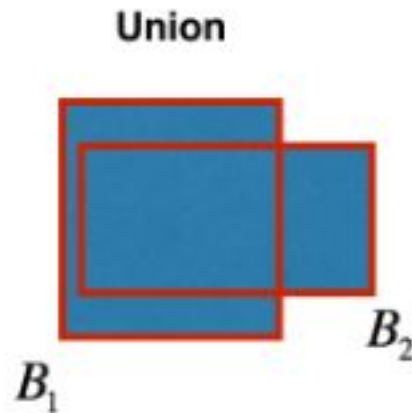
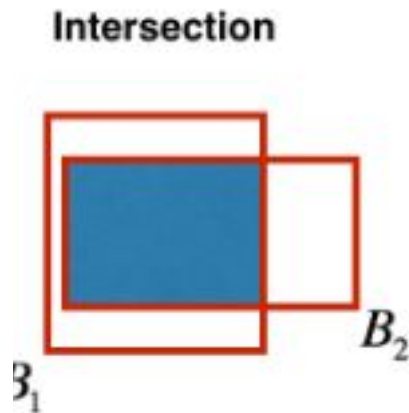
# Técnicas a emplear

- Técnica 2: Regresión del cuadro delimitador

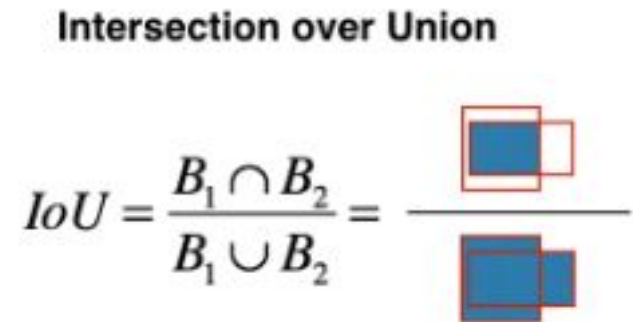


# Técnicas a emplear

- Técnica 3: Intersección sobre unión (IOU)



Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Intersection Area}}{\text{Union Area}}$$


# Técnicas a emplear

- Técnica 4: Supresión no máxima (Non-Max Suppression)



# Solución

**Se ingresa el comando en la consola:**

```
python3 yolo_video.py --input <PathVideoEntrada> --output  
<PathVideoSalida> --yolo yolo-coco
```

A screenshot of a terminal window with a dark background. The window title bar shows standard macOS window controls (red, yellow, green buttons) on the left, the text '-zsh' in the center, and 'Yolo-Vehicle-Counter' on the right. The terminal content shows a user prompt 'a.a.gallego@AMAC02FPGMSMD6M' followed by the command 'python3 yolo\_video.py --input input/3.mp4 --output output/3Count.avi --yolo yolo-coco'. The cursor is at the end of the command line.

```
a.a.gallego@AMAC02FPGMSMD6M Yolo-Vehicle-Counter % python3 yolo_video.py  
--input input/3.mp4 --output output/3Count.avi --yolo yolo-coco
```

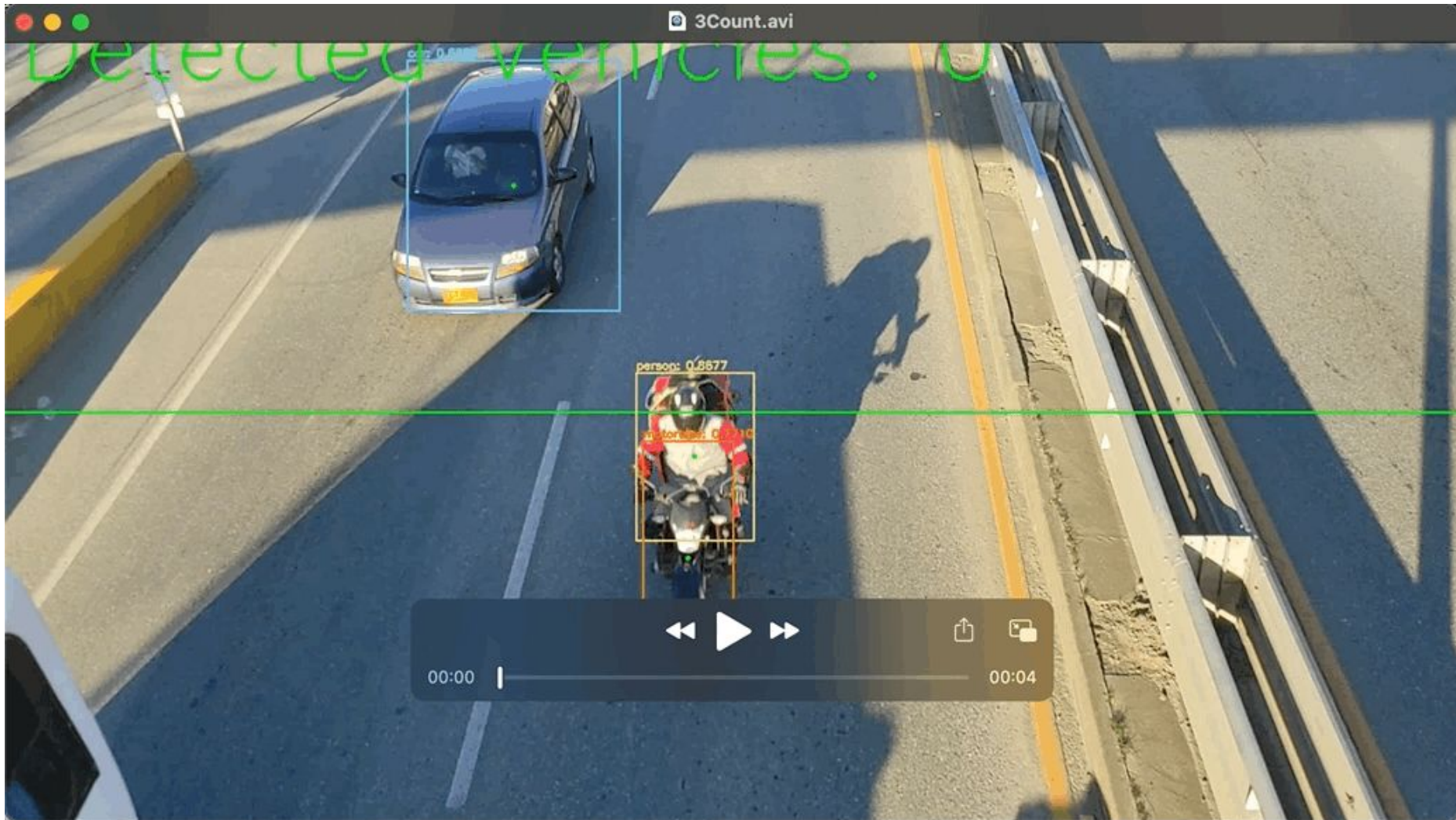
# Solución

- Se inicia el procesamiento del vídeo.



- Al finalizar, se obtiene el vídeo procesado con el conteo de vehículos en la ruta que se especificó al inicio.

# Solución





# Cronograma de actividades

Actividades	Tiempo						
	Semana 1	Semana 2	Semana 3	Semana 4	Semana 5	Semana 6	Semana 7
Elección de proyecto							
Descripción del problema							
Consultar YOLO							
Investigación, estado del arte							
Identificación técnicas							
Utilización técnicas con YOLO							
Toma de video							
Desarrollo y refinamiento							
Entrega final							

# Resultados, Líneas Futuras

En este proyecto, se realizó un proceso de investigación y exploración del algoritmo YOLO -You Only Look Once-. Se utilizaron los modelos entrenados por el algoritmo YOLO y su implementación para identificar objetos predefinidos, además, se incluyó en el algoritmo un contador de objetos, en este caso en particular, vehículos. Todo esto es posible gracias a las técnicas del procesamiento digital de imágenes. Cabe aclarar, que tanto la identificación como el conteo de dichos objetos no es óptimo, especialmente al momento de identificar motocicletas.

A futuro se podría mejorar y añadir nuevas funcionalidades como:

- Perfeccionar el algoritmo de conteo para que sea más preciso.
- Crear un aplicativo que permita subir un vídeo de una carretera con vehículos y retorne el total de vehículos que transitaron.
- Identificar la velocidad aproximada de un vehículo.
- Instalación de cámaras en puntos fijos, para el registro de información vehicular y así predecir flujos viales.
- Identificar elementos obligatorios, como el casco, en motociclistas.



# Referencias

- [1] Everingham, M., Van Gool, L., Williams, C. K. I. y col. El desafío PASCAL Visual Object Classes (VOC). Int J Comput Vis 88, 303–338 (2010). Recuperado el 07 de agosto de 2021, de: <https://doi.org/10.1007/s11263-009-0275-4>
  - [2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778). Recuperado el 09 de agosto de 2021, de: [https://openaccess.thecvf.com/content\\_cvpr\\_2016/papers/He\\_Deep\\_Residual\\_Learning\\_CVPR\\_2016\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2016/papers/He_Deep_Residual_Learning_CVPR_2016_paper.pdf)
  - [3] Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2117-2125). Recuperado el 09 de abril de 2021, de: [https://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/Lin\\_Feature\\_Pyramid\\_Networks\\_CVPR\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2017/papers/Lin_Feature_Pyramid_Networks_CVPR_2017_paper.pdf)
  - [4] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28, 91-99. Recuperado el 09 de agosto de 2021, de: <https://proceedings.neurips.cc/paper/2015/file/14bfa6bb14875e45bba028a21ed38046-Paper.pdf>
  - [5] Redmon, J., & Farhadi, A. (2017). YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7263-7271). Recuperado el 09 de agosto de 2021, de: [https://openaccess.thecvf.com/content\\_cvpr\\_2017/papers/Redmon\\_YOLO9000\\_Better\\_Faster\\_CVPR\\_2017\\_paper.pdf](https://openaccess.thecvf.com/content_cvpr_2017/papers/Redmon_YOLO9000_Better_Faster_CVPR_2017_paper.pdf)
  - [6] Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767. Recuperado el 09 de agosto de 2021, de: <https://arxiv.org/pdf/1804.02767.pdf>
  - [7] Apple detection during different growth stages in orchards using the improved YOLO-V3 model. Recuperado el 09 de agosto de 2021, de: <https://www.sciencedirect.com/science/article/abs/pii/S016816991831528X>
-