

## Taller Final

### Visión Computacional

---

#### 1 . Objetivo general

Diseñar e implementar una solución de **visión computacional basada en deep learning** que genere un beneficio social concreto y medible. Cada grupo deberá:

- **Definir claramente el problema** y explicar de qué manera su propuesta crea valor social.
- **Desarrollar una arquitectura compuesta** por, al menos, **dos tareas de visión** (p. ej. detección + segmentación, segmentación + tracking, matching etc.). La clasificación puede añadirse como tercera tarea opcional.
- **Entrenar o reentrenar** al menos **un modelo/componente** del sistema con un **conjunto de datos propio**.

**Nota:** Si se utiliza un único modelo que ya resuelve todo el flujo de forma monolítica, la calificación será mínima: lo que interesa es la **orquestración creativa** de varias técnicas.

---

#### 2 . Selección del problema

Puede continuar el proyecto del segundo corte (siempre que el beneficio social sea evidente) o elegir uno de los tópicos sugeridos. También se aceptan propuestas originales, incluidas problemáticas dentro de la universidad.

Ejemplos sugeridos	Descripción
Monitoreo de ocupación en transporte público	Estimar niveles de saturación en buses o estaciones para optimizar frecuencias o alertar a usuarios.
Detección de comportamientos peligrosos en motociclistas	Identificar uso inadecuado de casco, exceso de pasajeros o maniobras de riesgo.
Detección y seguimiento de deforestación	Cuantificar áreas deforestadas mediante imágenes satelitales

---

#### 3. Requisitos técnicos mínimos

**Modelado:**

1. Uso predominante de **redes neuronales profundas**.
2. **Combinación secuencial o paralela** de  $\geq 2$  tareas de visión (detección, segmentación, tracking, matching, etc.).

#### Datos:

1. Reentrenamiento de al menos **un modelo** con **dataset propio**.
2. Evitar datasets idénticos entre grupos: la repetición conllevará **calificación 1.0**.

**Entorno de ejecución:** Todo el pipeline debe poder correrse en **Google Colab con GPU**.

**Video:** El procesamiento de video otorga **bonificación** (no obligatorio).

---

#### 4. Entregables

Entregable	Detalles
Repositorio GitHub (se habilitará por el docente)	<ul style="list-style-type: none"><li>• Código limpio y comentado (scripts + notebooks)</li><li>• run_pipeline.py que ejecute de principio a fin</li><li>• Archivo de dependencias (requirements.txt ó environment.yml)</li><li>• Carpeta de datos claramente identificada</li></ul>
Reporte técnico (README.md)	<ol style="list-style-type: none"><li>1. Resumen del problema y su impacto social</li><li>2. Descripción de la arquitectura y justificación de decisiones</li><li>3. Detalle de los datasets (propios y externos)</li><li>4. Métricas empleadas y discusión de resultados</li><li>5. Lecciones aprendidas y trabajo futuro</li></ol>

---

#### 5. Criterios de evaluación

Criterio	%
Claridad del planteamiento social y justificación	1
Diseño de la arquitectura (orquestración de tareas, creatividad, documentación)	3
Presentación de los scripts y artefactos (claridad, coherencia)	1

*El día de la sustentación al docente (no es a todo el salón), este preguntará por el entendimiento teórico de las arquitecturas/modelos utilizadas en el proyecto. Se harán hasta 2 preguntas por integrante del grupo, que se contestarán de forma individual. Por cada pregunta mal contestada, la nota se reduce en 1.5.*

*Integrante que no esté el día de la presentación, tiene como nota cero.*

*Proyectos excepcionalmente innovadores pueden tener un impacto muuuuuuy positivo en el parcial final (es a discreción del docente).*

---

## 6. Penalizaciones

- **Repositorio mal organizado:** hasta  $-2.0$  puntos. Asegúrese de hacer el
  - **Incumplimiento del requisito de dataset propio:** hasta  $-3.0$  puntos.
- 

## 7. Entrega y cronograma

- **Grupos de a dos**
- **Fecha límite de subida al repositorio:** *[25-05-2025 – 23:59 h]*
- **Demostración en clase** (max 5 min de presentación + 5 min de preguntas): *26/30-05-2025.*  
*Si presenta el 26 o 30 es por sorteo.*

**¡Éxitos en el desarrollo!**