



Facultad de Ciencias Exactas, Ingeniería y Agrimensura

Tecnicatura Universitaria en Inteligencia Artificial

Trabajo Final: Computer Vision

Integrante: Mateo Rovere

Introducción:

Este es el informe para el Trabajo Práctico Final de Visión por Computadora en el cual fue trabajado una detección de cartas españolas que serán utilizadas para ver si hay puntos para envío, y en caso de que si haya, cual es el puntaje.

El envío es un canto que se utiliza para desafiar al equipo contrario a mostrar la suma de las cartas de mayor valor que tienen en su mano.

El envío se obtiene de la suma de dos cartas del mismo palo. A esta suma se le agregan 20 puntos.

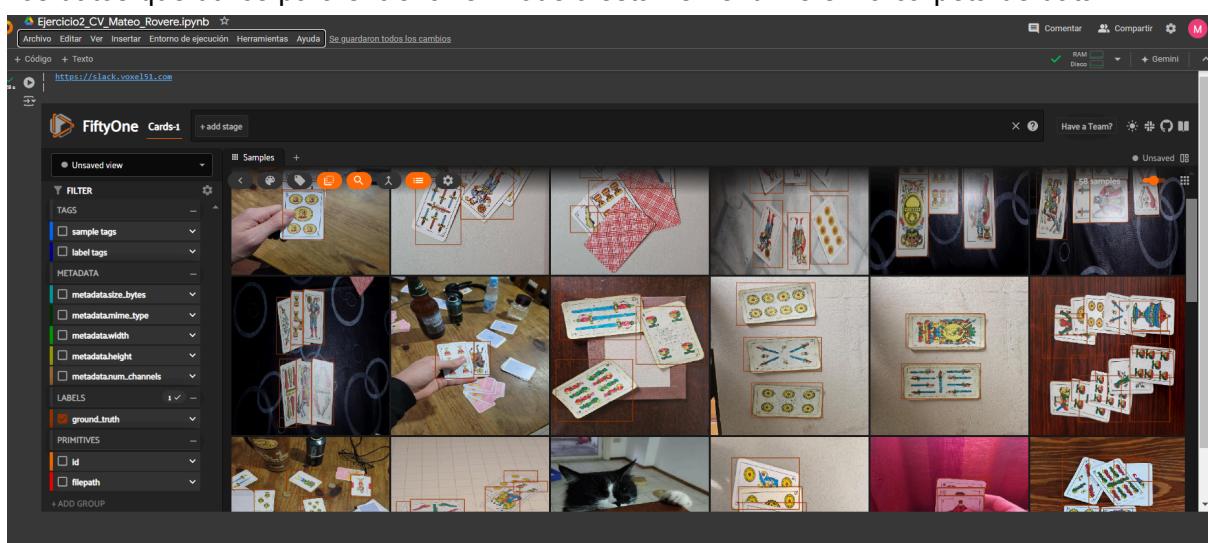
Por ejemplo, si tienes un 5 y un 6 del mismo palo, la suma sería 11. Agregando los 20 puntos, tendrías un total de 31 puntos en el envío.

El puntaje mínimo es 0 (tres cartas de distinto palo), y el máximo es 33 (un 6 y un 7 del mismo palo).

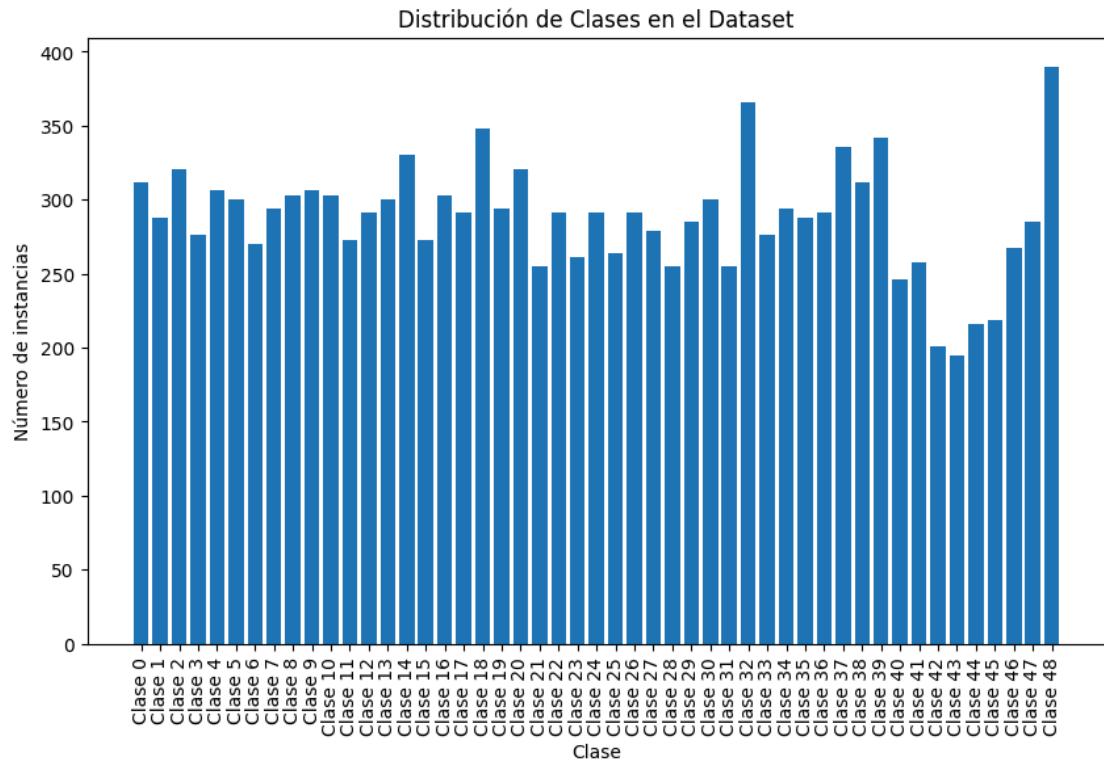
Proceso de evaluación de datos:

Para poder decidir qué datos me parecen pertinentes y cuales no, decidí ver uno por uno las carpetas (con la ayuda de FiftyOne) y sus anotaciones para darme cuenta si esa persona etiquetó los datos de manera similar a la mía (si bien había un consenso de cómo etiquetar los datos, hay casos en los que no se cumple), y con esas arme un dataset propio.

Los datos que utilice para entrenar el modelo están en el drive en la carpeta de data



Luego quise ver que tan balanceados estaban los datos entonces realice este gráfico:



Se puede ver que en general no hay un gran desbalance de clases pero sí se puede percibir que la carta que más aparece es el comodín que es la clase 48, y hay pocas imágenes con los 8 y los 9 (clase 40 a la 47)

Proceso de trabajo realizado:

Luego de obtener los datos, los añadí en Roboflow que es un servicio para proyectos de computer vision muy útil en el cual se encargó de hacer el aumentado de datos, y de separar en train y val (que utilice una proporción de 0.9 de entrenamiento). Luego utilice esos datos en un archivo .ipynb para hacer el entrenamiento con Yolov10x dado que tiene una gran mejora comparado a sus antecesoras en velocidad de inferencia. El entrenamiento fue realizado en Kaggle dado que me brindaba una GPU P100, que es mejor que la que da Colab de manera gratuita.

Se separaron el 90% de las imágenes para entrenar y el 10% para validación, pero luego solo se implemento aumentado de datos a la parte de entrenamiento con la ayuda de Roboflow quedando 2361 imágenes de entrenamiento y 58 de validación. Para el testeo decidí hacerlo con video de manera manual para no perder datos de imágenes que podrían usarse para entrenar

El aumentado de datos que se llevó a cabo incluía:

Flip: Horizontal, Vertical

90° Rotate: Clockwise, Counter-Clockwise, Upside Down

Saturation: Between -12% and +12%

Brightness: Between -10% and +10%

Exposure: Between -5% and +5%

Luego para el análisis de las métricas me dio este resultado:

Class	Images	Instances	Box (P)	R	mAP50	mAP50-95
all	58	304	0.815	0.802	0.854	0.699
10B	58	3	0.793	0.667	0.863	0.758
10C	58	9	0.741	1	0.926	0.815
10E	58	2	0.414	0.5	0.695	0.615
10O	58	5	0.716	0.8	0.756	0.603
11B	58	7	0.898	0.857	0.904	0.781
11C	58	4	1	0.94	0.995	0.796
11E	58	8	0.647	0.625	0.689	0.618
11O	58	2	0.382	1	0.995	0.797
12B	58	7	1	0.747	0.877	0.805
12C	58	9	0.908	0.889	0.968	0.874
12E	58	4	0.758	0.75	0.888	0.765
12O	58	8	1	0.836	0.888	0.677
1B	58	9	0.848	0.626	0.744	0.584
1C	58	10	0.818	0.7	0.766	0.646
1E	58	8	0.975	1	0.995	0.911
1O	58	3	0.93	1	0.995	0.853
2B	58	4	0.806	0.5	0.592	0.409
2C	58	7	0.951	0.714	0.902	0.644
2E	58	6	0.909	1	0.995	0.917
2O	58	4	0.588	0.381	0.564	0.397
3B	58	5	0.636	0.8	0.796	0.736
3C	58	3	0.764	0.667	0.741	0.547
3E	58	6	0.927	0.833	0.942	0.718
3O	58	6	0.743	1	0.924	0.664
4B	58	7	1	0.519	0.66	0.379
4C	58	5	0.649	0.746	0.773	0.566
4E	58	11	0.824	0.856	0.892	0.773
4O	58	3	0.586	0.954	0.753	0.312
5B	58	6	0.985	1	0.995	0.763
5C	58	4	0.547	0.75	0.685	0.585
5E	58	8	0.73	0.75	0.885	0.754
5O	58	11	0.904	0.856	0.946	0.803
6B	58	3	0.704	1	0.995	0.852
6C	58	10	1	0.757	0.896	0.62
6E	58	8	0.87	0.837	0.931	0.765
6O	58	5	0.902	1	0.995	0.818
7B	58	7	1	0.81	0.92	0.833
7C	58	10	0.711	0.9	0.861	0.751
7E	58	6	0.76	1	0.948	0.817
7O	58	10	1	0.782	0.844	0.732
8B	58	3	0.725	1	0.913	0.765
8C	58	7	0.685	0.628	0.646	0.466
8E	58	6	1	0.785	0.942	0.826
8O	58	8	0.855	0.741	0.773	0.664
9B	58	6	0.827	0.802	0.809	0.601
9C	58	6	0.842	0.667	0.788	0.552
9E	58	3	0.847	1	0.995	0.907
9O	58	7	0.849	0.714	0.792	0.721
J	58	5	1	0.593	0.824	0.672

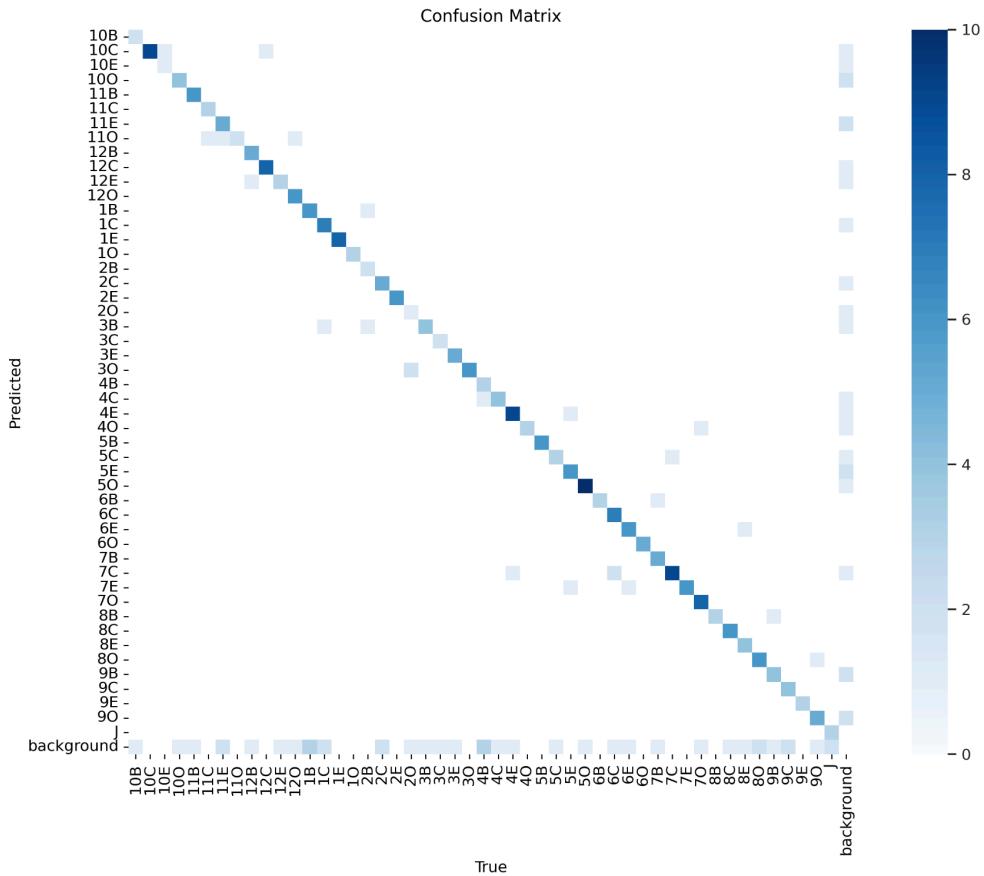
Con un map50-95 promedio de 0.699 que nos indica un buen desempeño del modelo.

Ejemplo de imagen de cartas con superposición de ground truth y bounding box de la predicción:

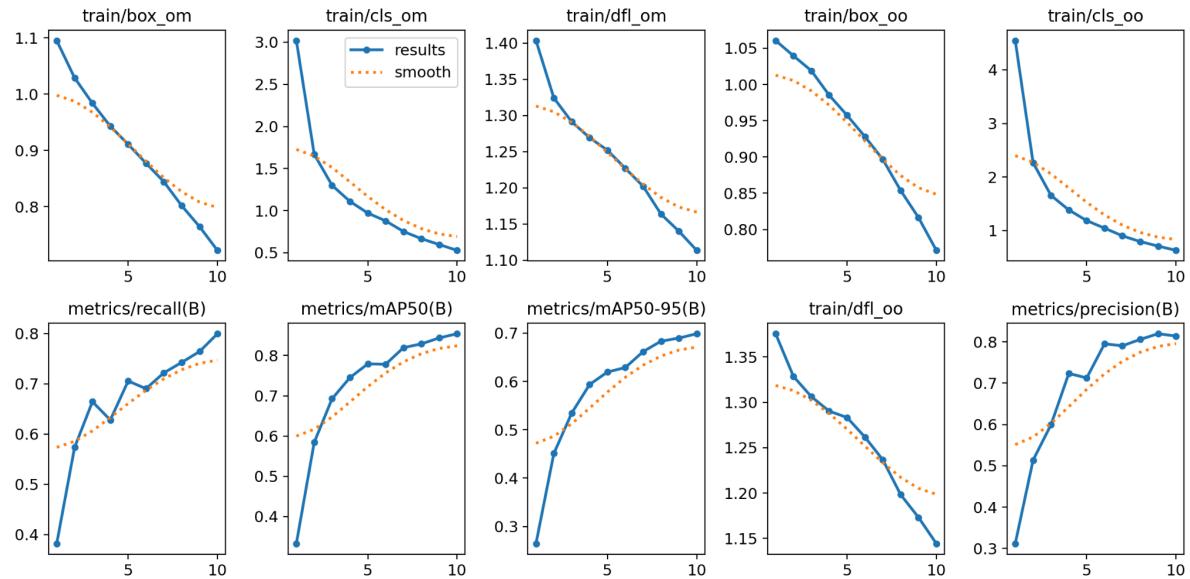


Se puede ver como en esta imagen consigue encontrar el 3 y 4 de copas con facilidad pero tiene problemas con el 6 de copas, se puede observar que las cartas encontradas tienen una muy buena detección, sin mucho fondo detectado como carta, y también se aprecia que no hay ninguna carta detectada donde es fondo. Para aclaración, los recuadros azules son los ground truth y los rojos son las predicciones.

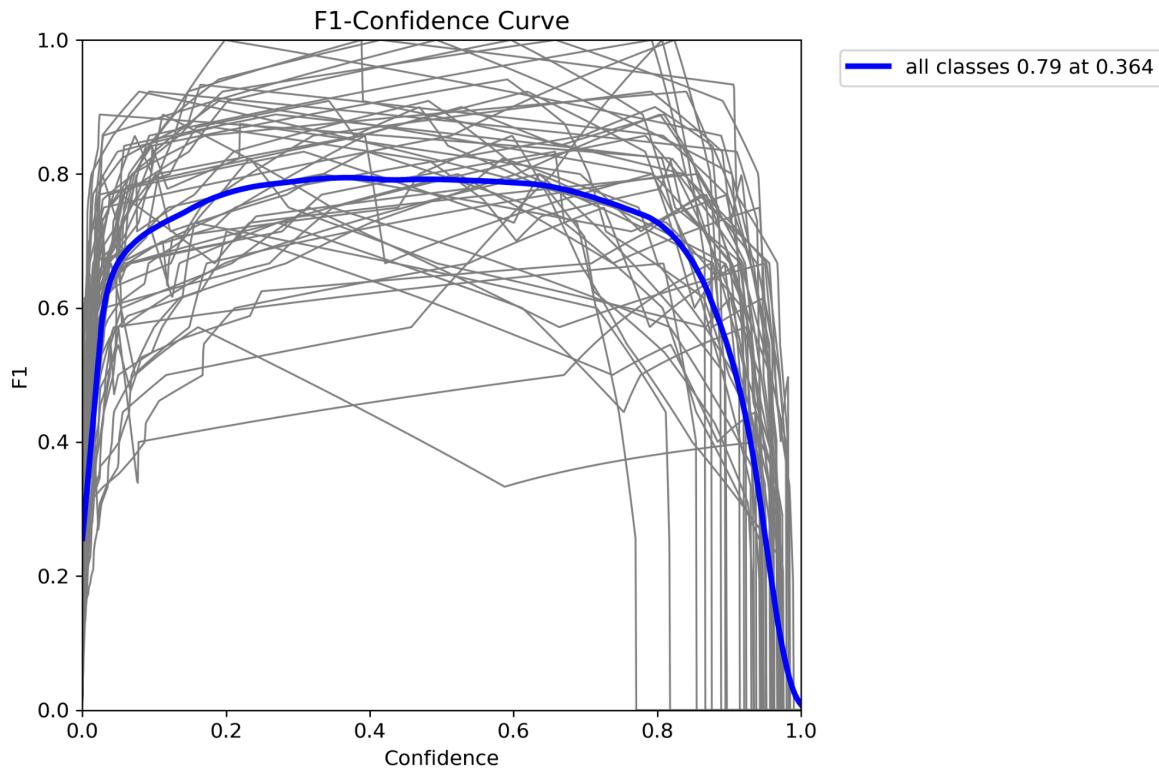
Y viendo la matriz de confusión podemos ver que clasifica bastante bien, pero el error más común es que hay veces que confunde una carta por background, osea que no la detecta



Con el resto de las métricas podemos observar como la precisión y el recall están cerca del 0.8:



También podemos ver que el F1 más alto para el promedio de las clases lo encontramos con una confianza de 0.364 con un valor de 0.79:



Conclusiones:

- Clases con Buen Desempeño: Las clases como 1O, 1E, 11C, 12O y 7B tienen buenos valores de mAP50, mAP50-95, Precision y Recall, mostrando un desempeño sólido en términos de detección.
- Clases con Bajo Desempeño: Los 8 y los 9 muestran bajos valores en precisión y recall, indicando que el modelo tiene dificultades para detectar estas clases con precisión dado la menor cantidad de imágenes
- .
- General: El modelo es bueno en la detección de muchas clases, pero hay margen para mejorar especialmente en las clases con baja precisión y recall que se puede solucionar dando más épocas al modelo y teniendo más imágenes en el dataset, también estaría bueno experimentar con batch size mayor, pero para eso no tenía los recursos suficientes.

Inferencia de imagen y envío sobre imágenes que brindó la cátedra para hacer testeo del modelo:



```
"IMG_20240630_174105714.jpg": {
    "total_cards": 1,
    "cards": {
        "E": [],
        "C": [],
        "B": [
            8
        ],
        "O": []
    },
    "points": 0,
    "figure": "N/A"
}
```

} ,

Cabe aclarar que puse una validación para que únicamente diga los puntos en caso de que existan 2 cartas en la mesa, en caso de ser más o menos no aparecen los puntos del envido.



```
"IMG_20240630_173822184_HDR.jpg": {
    "total_cards": 3,
    "cards": {
        "E": [
            7
        ],
        "C": [],
        "B": [
            6,
            2
        ],
        "O": []
    }
}
```

```
},
"points": 28,
"figure": "B"
}
```

En este caso como cumple la condición podemos ver que si aparecen los puntos del envido aunque clasifica erróneamente un 3 de basto por un 2 de basto.