

Tecnológico de Monterrey

Proyecto Integrador

Avance 1. Análisis exploratorio de datos

Marcela Alejandra Rosales Jiménez – A01032022

José Antonio Mendoza Castro – A01794067

Profa. Dra. Grettel Barceló Alonso

Prof. Dr. Luis Eduardo Falcón Morales

Profa. Verónica Sandra Guzmán de Valle

29 de septiembre de 2024

Índice

Análisis exploratorio de datos	3
Repositorio.....	7
Bibliografía	7
Anexos	7

Análisis exploratorio de datos

Se utilizará el dataset *Woven Traffic Safety Dataset* (WTS) que contiene cerca de 1,200 videos con eventos de tráfico enfocados en el peatón, adicional a videos externos que fueron incluidos en la base de datos.

En el dataset se muestran varias escenas actuadas de lo que es una interacción vial que involucra uno o varios peatones y carros. Los 1,200 videos se componen tanto de video tomado desde el interior de un carro que el dataset llama *vehicle_view* y también videos desde un ángulo superior que le llama *overhead_view*. Algunos de los videos de *overhead_view* son de la misma escena, pero desde diferentes ángulos del conflicto vial.

Resumiendo, datos generales sobre de la base de datos:

- Tiene cerca de 1200 segmentos de videos (adicional a videos externos incluidos en la base)
- El tamaño de la imagen de los videos es de 1920x1080
- Contiene archivos .mp4, .json y .csv

```
In [46]: def get_all_files(files_path):
        files_path = pathlib.Path(files_path)
        listed_elements = list(files_path.rglob("*"))
        listed_files = [e for e in listed_elements if not e.is_dir()]
        return listed_files

        def get_listed_files_with_types(listed_files):
            files_types = {}
            for ff in listed_files:
                ext = ff.suffix
                if not files_types.get(ext):
                    files_types[ext] = []
                files_types[ext].append(str(ff))
            return files_types

        listed_files = get_all_files(f"{dataset_path}")
        print(f"Cantidad de archivos en total = {len(listed_files)}")

        files_by_ext = get_listed_files_with_types(listed_files)

        for k, v in files_by_ext.items():
            print(f" - Extension {k} tiene {len(v)}")
```

```
Cantidad de archivos en total = 14126
- Extension .csv tiene 1
- Extension .zip tiene 4
- Extension tiene 1
- Extension .DS_Store tiene 1
- Extension .json tiene 9907
- Extension .mp4 tiene 4212
```

Podemos ver que dentro de el zip del dataset hay 4,212 videos.

Figura 1. Características generales de la base de datos

Ya que el enfoque de este proyecto es el de analizar las escenas desde una perspectiva de espectador y no centrándonos en el vehículo, se ha decidido descartar los videos contenidos en los folders `vehicle_view` y `wts_dataset_zip/external`. Se consideran videos tanto de train como de validation, por lo que en total se cuentan con 600 videos con sus anotaciones.

De los datos seleccionados sobresale:

- La mayoría de los videos de `overhead_view` tienen un conteo de 2000 frames, con algunos outliers de 7700 frames. Ver la figura 1.
- Los videos de `overhead_view` son de 30 frames per second
- Los videos duran entre 1 y 4 minutos.

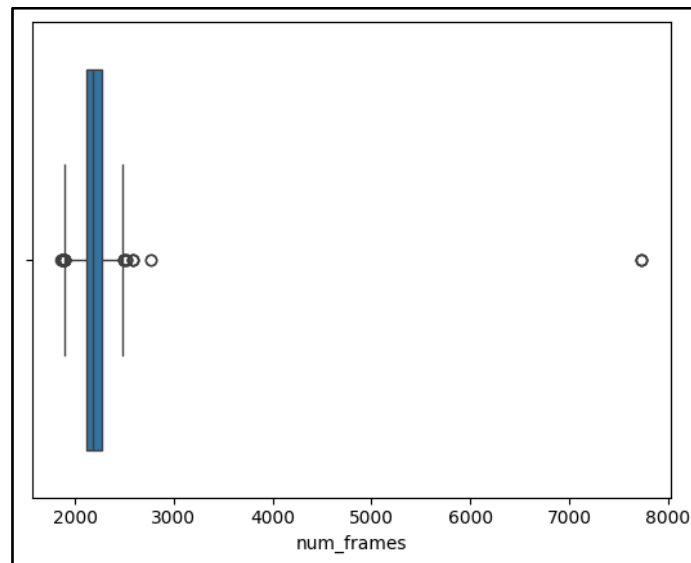


Figura 2. Gráfica de caja para mostrar la distribución del número de frames en los videos de `overhead_view`

Adicional obtuvimos los segundos de duración por video, se calculó la media en segundos de duración de los videos seleccionados a partir de una muestra aleatoria, teniendo 52 segundos.

```
[ ] list_seconds = []

for video in video_sample:
    blob_tmp = bucket.blob(video)
    blob_tmp.download_to_filename('/content/' + video.split('/')[-1])
    cap = cv2.VideoCapture(video.split('/')[-1])
    fps = cap.get(cv2.CAP_PROP_FPS)
    totalNoFrames = cap.get(cv2.CAP_PROP_FRAME_COUNT)
    durationInSeconds = totalNoFrames // fps
    list_seconds.append(durationInSeconds)

[ ] print('Promedio de duracion de los videos en segundos: ', sum(list_seconds) / len(list_seconds))
```


 Promedio de duracion de los videos en segundos: 52.6

Figura 3. Cálculo de la media de duración de los videos en segundos

El dataset cuenta con un archivo de annotation para cada video. En estas anotaciones se cubre el comportamiento del peatón en 5 fases (pre-reconocimiento, reconocimiento, juicio, acción y evasión). En cada fase se hace una descripción textual del comportamiento del peatón para cada segmento. En la figura 2 se muestra un ejemplo de las primeras líneas de un archivo de anotaciones.

```
files_overhead_view_by_type['train'][0]

'dataset/wts_dataset_zip/videos/train/20231006_15_CN28_T1/overhead_view/20231006_15_CN28_T1_192.168.0.13_4.mp4'

! cat dataset/wts_dataset_zip/annotations/caption/train/20231006_15_CN28_T1/overhead_view/20231006_15_CN28_T1_caption.json

{
  "id": 1720,
  "overhead_videos": [
    "20231006_15_CN28_T1_192.168.0.11_1.mp4",
    "20231006_15_CN28_T1_192.168.0.12_3.mp4",
    "20231006_15_CN28_T1_192.168.0.13_4.mp4",
    "20231006_15_CN28_T1_192.168.0.28_2.mp4"
  ],
  "event_phase": [
    {
      "labels": [
        "4"
      ],
      "caption_pedestrian": "The pedestrian is a male in his 30s with a height of 170 cm. He is wearing a black T-shirt and black slacks. It is a clear and bright day with the road surface being dry and level. The pedestrian is walking in front of the vehicle, in the opposite direction, and on the left side. He is relatively close to the vehicle. The pedestrian notices the vehicle approaching and traveling on a residential road with two-way traffic. The pedestrian is slowly walking in a car lane. There is no sidewalk on either side, and the roadside strip is also absent. The pedestrian's general action is a collision. Overall, the pedestrian is in an environment with typical traffic volume, on a residential road with asphalt. The pedestrian's movements and the surrounding conditions contribute to a potentially dangerous situation.",
      "caption_vehicle": "The vehicle was positioned on the left side of a pedestrian, at a close distance. The pedestrian was clearly visible within the vehicle's field of view. Unexpectedly, the vehicle collided with the pedestrian, causing both parties to come to a halt. The vehicle was stationary at the time of the collision, with a speed of 0 km/h. Meanwhile, in the surrounding environment, the pedestrian was described as a male in his 30s, of approximately 170 cm in height. He was wearing a black T-shirt for his upper body and black slacks for his lower body. The weather conditions were clear, with bright brightness. The road surface was dry and level, consisting of asphalt. The road was a residential road with two-way traffic and no sidewalks on both sides. Additionally, there were no roadside strips. The event occurred in the usual traffic volume of the area.",
      "start_time": "30.803",
      "end_time": "33.681"
    },
    {
      "labels": [
        "3"
      ],
      "caption_pedestrian": "The pedestrian, a male in his 30s dressed in a black T-shirt and slacks, was spotted diagonally to the left in front of the vehicle on a residential road. His body was oriented in the opposite direction of the vehicle, and his line of sight was focused ahead in the direction of travel. With clear and bright weather conditions, the road surface was dry and level, providing optimal visibility and ease of movement. Despite the usual traffic volume on the two-way road, the pedestrian was closely watching the vehicle as he slowly proceeded straight ahead, traveling within the car lane. Both the pedestrian's appearance and actions indicated a cautious and aware demeanor. The surrounding environment, characterized by asphalt road and no sidewalks on both sides, contributed to the pedestrian's position and behavior."
    }
  ]
}
```

Figura 3. Ejemplo de un archivo JSON con anotaciones sobre un video

Consideramos ejecutar algunos videos de forma aleatoria para entender los tipos de escenarios contenidos en los datos seccionados.

```
[7] def play_random_video(list_video):
    """
    Funcion que ejecuta un video aleatorio desde un bucket en GCP.
    """

    random_number = randrange(1, len(list_video)) - 1
    blob_tmp = bucket.blob(blobs[random_number])
    blob_tmp.download_to_filename('/content/' + blobs[random_number].split('/')[1])

    video_path = blobs[random_number].split('/')[1]

    video = open(video_path, 'rb').read()
    video_data_url = 'data:video/mp4;base64,' + b64encode(video).decode()

    print('Video Path: ', video_path, 'Random number: ', random_number)

    return HTML(f"""
    <video width="640" height="480" controls>
    <source src="{video_data_url}" type="video/mp4">
    </video>
    """)
```

Figura 4. Función para reproducir un video de forma aleatoria

De los videos reproducidos se observaron las perspectivas que tienen los peatones, en donde se pueden tener escenarios en los que están cruzando la calle, en situaciones en las que el peatón puede estar atento a su contexto o no y donde algún o más autos intervienen, dando la vuelta en una esquina o cambiando de velocidad, por ejemplo.

```
✓ [14] play_random_video(blobs)
13s
📺 Video Path: 20231013_114328_normal_192.168.0.17_1_event_1.mp4 Random number: 459
```



Figura 5. Reproducción de un video de la base WTS

En general la primera exploración de datos nos da un panorama general de los elementos que conforman la base, si bien no se puede realizar una exploración directa como en una base de datos estructurada con la aplicación de algunas técnicas de LLMs, NLP y ML podremos obtener más información de la base de datos.

Repositorio

https://github.com/marcelarosalesj/mna_proyecto_integrador_equipo13

Bibliografía

Wang, S., Anastasiu, D. C., Tang, Z., Chang, M.-C., Yao, Y., Zheng, L., Rahman, M. S., Arya, M. S., Sharma, A., Chakraborty, P., Prajapati, S., Kong, Q., Kobori, N., Gochoo, M., Otgonbold, M.-E., Alnajjar, F., Batnasan, G., Chen, P.-Y., Hsieh, J.-W., Wu, X., Pusegaonkar, S. S., Wang, Y., Biswas, S., & Chellappa, R. (2024). The 8th AI City Challenge. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Kong, Q., Kawana, Y., Saini, R., Kumar, A., Pan, J., Gu, T., Ozao, Y., Opra, B., Anastasiu, D. C., Sato, Y., & Kobori, N. (2024). WTS: A pedestrian-centric traffic video dataset for fine-grained spatial-temporal understanding. arXiv. <https://arxiv.org/abs/2407.15350>

Duan, Z., Cheng, H., Xu, D., Wu, X., Zhang, X., Ye, X., & Xie, Z. (2024). CityLLaVA: Efficient Fine-Tuning for VLMs in City Scenario. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

Tecnológico de Monterrey. (n.d.). Futuro Ciudades. Retrieved September 22, 2024, from <https://futurociudades.tec.mx/es>

Hugging Face. (n.d.). Learn how vision-language pretraining. Retrieved from https://huggingface.co/blog/vision_language_pretraining#:~:text=Learn%20how%20vision-language

Instituto de Políticas para el Transporte y el Desarrollo. (2020, August 25). Análisis de conflictos viales: Metodología y resultados en el entorno escolar de BINE Puebla. Retrieved from <https://mexico.itdp.org/2020/08/25/analisis-de-conflictos-viales-metodologia-y-resultados-en-el-entorno-escolar-de-bine-puebla>

Instituto Nacional de Estadística y Geografía. (n.d.). Accidentes de tránsito terrestre. INEGI. Retrieved September 22, 2024, from <https://www.inegi.org.mx/temas/accidentes/>

Anexos

Para esta entrega no se consideró información adicional.