Herramienta de anotación de video automática basada en aprendizaje profundo para automóviles autónomos

NSManikanda,
TIFAC-CORE en Infotrónica Automotriz,
Instituto de Tecnología de Vellore,
Katpadi, Vellore, Tamilnadu, India-632014
nsmanikandan@vit.ac.in

K. Ganesan,
TIFAC-CORE en Infotrónica Automotriz,
Instituto de Tecnología de Vellore, Katpadi,
Vellore, Tamilnadu, India-632014
kganesan@vit.ac.in

Abstracto
En un automóvil autónomo, la detección de objeciones, la clasificación de objetos, la detección de carriles y el seguimiento de objetos se consideran módulos cruciales. En los últimos tiempos, mediante el vídeo en tiempo real se quiere narrar la escena captada por la cámara instalada en nuestro vehículo. Para implementar de manera efectiva esta tarea, se utilizan ampliamente técnicas de aprendizaje profundo y herramientas de anotación automática de video. En el presente documento, comparamos las diversas técnicas disponibles para cada módulo y elegimos el mejor algoritmo entre ellas mediante el uso de métricas apropiadas. Para la detección de objetos, se consideran YOLO y Retinanet-50 y se elige el mejor en función de la precisión promedio promedio (mAP). Para la clasificación de objetos, consideramos VGG-19 y Resnet-50 y seleccionamos el mejor algoritmo en función de la baja tasa de error y la buena precisión. Para la detección de carril, Se comparan los algoritmos 'Finding Lane Line' de Udacity y LaneNet basados en aprendizaje profundo y se elige el mejor que puede identificar con precisión el carril dado para la implementación. En lo que respecta al seguimiento de objetos, comparamos el algoritmo de "Detección y seguimiento de objetos" de Udacity y el algoritmo de clasificación profunda basado en aprendizaje profundo. En función de la precisión del seguimiento del mismo objeto en muchos fotogramas y de la predicción del movimiento de los objetos, se elige el mejor algoritmo. Nuestra herramienta automática de anotación de video tiene una precisión del 83 % en comparación con un anotador humano. Consideramos un video con 530 fotogramas cada uno de resolución 1035 x 1800 píxeles. En promedio, cada cuadro tenía alrededor de 15 objetos. Nuestra herramienta de anotación consumió 43 minutos en un sistema basado en CPU y 2,58 minutos en un sistema basado en GPU de nivel medio para procesar los cuatro módulos. Pero el mismo video tomó casi 3060 minutos para que un anotador humano narrara la escena en el video dado. Por lo tanto, afirmamos que nuestra herramienta de anotación de video automática propuesta es razonablemente rápida (alrededor de 1200 veces en un sistema de GPU) y precisa.

Palabras clave: anotación automática, aprendizaje profundo, clasificación de objetos, detección de objetos, detección de carriles y seguimiento de objetos

Introducción:

La organización mundial de la salud ha encuestado a 180 países del mundo y ha informado que 1,25 millones de personas mueren cada año debido a accidentes de tráfico. La tasa de mortalidad es alta en los países de bajos ingresos [1]. Una de las formas en que podemos reducir esta tasa de mortalidad es usar automóviles sin conductor. En una encuesta reciente, el 69% de los encuestados informaron que los automóviles sin conductor son más seguros que los automóviles conducidos por humanos [2]. En un automóvil sin conductor, las funciones de control críticas para la seguridad, como la dirección, la aceleración y el frenado, se realizan sin la interferencia del conductor [3]. Hay muchos niveles de automatización de los coches sin conductor. Uno de los niveles es utilizar la visión artificial. La detección y clasificación de objetos son los problemas cruciales en la visión artificial. Los sistemas de detección y clasificación detectan y clasifican los distintos objetos que se encuentran en la vía, especialmente vehículos, peatones y objetos estacionarios en el costado de la carretera, como señales de tráfico, letreros, postes de luz. Para el desarrollo de modelos de detección y clasificación, se necesitan conjuntos de datos de entrenamiento en tiempo real. Pero estos conjuntos de datos utilizan una gran cantidad de imágenes. Los objetos presentes en este conjunto de datos son anotados manualmente por seres humanos.. Durante el proceso de anotación, dibujan cuadros delimitadores alrededor de los objetos identificados y también narran (almacenan) las propiedades de estos objetos. los anotadores generalmente usa las herramientas de anotación de código abierto (manual) [4].

Con estas herramientas, se pueden crear cuadros delimitadores para la localización de objetos, dibujar polígonos para la segmentación de objetos y agregar etiquetas usando texto en las regiones elegidas. Los datos anotados se almacenan en muchos formatos, como texto XML, JSON, YOLO[5], ILSVRC[6], etc. La anotación manual no solo es costosa sino que también requiere mucho tiempo. Por ejemplo, la base de datos de detección de objetos de ILSVRC [6] necesita unos 42 segundos para dibujar un cuadro delimitador alrededor de un objeto [7]. Para hacer que este proceso de anotación de cuadro delimitador sea más económico y preciso, los investigadores adoptan dos estrategias diferentes. Son métodos de anotación semiautomáticos y totalmente automáticos.

Durante el desarrollo de semiautomático herramienta de anotación Dim P. Papadopoulos et. al[8]. han reducido el tiempo de anotación mediante el uso de una arquitectura de anotación de clic central y en su herramienta pidieron a los anotadores que hicieran clic en el centro del objeto presente en una imagen. Se descubrió que este método era rápido y también reducía el tiempo de anotación entre 9 y 18 veces. Adithya Subramanian et al[9]. han presentado una nueva metodología para anotar rápidamente los datos utilizando técnicas de supervisión de clics y detección de objetos jerárquicos. Utilizaron el método semiautomático. La tarea de las anotaciones se dividió entre el ser humano y una red neuronal. El

El marco propuesto por ellos fue 3-4 veces más rápido que los métodos de anotación manual estándar.

Dim P. Papadopoulos et. al[10]. han propuesto una herramienta de anotación totalmente automática en la que los objetos se detectan mediante un algoritmo de aprendizaje. Los anotadores humanos simplemente verificaban los cuadros delimitadores. Su método redujo el tiempo de anotación entre 6 y 9 veces. . Zhujun Xiao et.al[11]. han diseñado una herramienta de generación de imágenes con autoanotación mediante la combinación de la cámara con un localizador inalámbrico pasivo. Los módulos de detección de peatones y vehículos se utilizaron como ejemplos. Han demostrado la viabilidad, los beneficios y los desafíos de un sistema automático de anotación de imágenes.

ciclomotor, motociclista, ciclista, otro vehículo de dos ruedas y anodino. El objeto de dos ruedas detectado se caracteriza además por diez propiedades tales como oclusión, oclusión de la cabeza, oclusión de los pies, dirección, movimiento, asignación de carril, rotación, pose, iluminación y medición del cuadro delimitador (tamaño). La Tabla 3 clasifica a un ser humano detectado como peatón y anodino. El peatón detectado se caracteriza además por nueve propiedades, como la oclusión, la oclusión de la cabeza, la oclusión de los pies, la dirección, el movimiento, la altura, la pose extraña, la iluminación y la medida del cuadro delimitador (tamaño).

Tabla 1: Propiedades del vehículo

Objeto tipo	Oclusión	Abajo Oclusión	Dirección	Movimienot	carril asignación	carril cambiar detección	Rotación	Pose	Encendiendo	Tamaño
auto/ autobús/ camión/ otro-vehículo/ no-describir	ninguno/ parcial/ lleno	verdadero/ FALSO	anterior/ venidero	Moviente/ estacionario/ estacionado	desconocido/-2/-1/0/+1/+2	verdadero/ FALSO	importante/ irrelevante	trasero/ abajo a la derecha/ trasera izquierda/ frente/ frente derecho/ delantero izquierdo/ lado	normal/ desenfocar/ destello	marta, mio, máximo, maximo

Tabla 2: Propiedades de dos ruedas

Tipo de objeto	Oclusión	Cabeza Oclusión	Pies oclusión	Dirección	Movimienot	carril asignación ent	Rotación	Pose	Encendiendo	Tamaño
motociclista/ motociclista/ ciclista/ otros dos-rodador/ mediocre	ninguno/ parcial/ lleno	verdadero/ FALSO	verdadero/ FALSO	anterior/ venidero	Moviente/ estacionario/ estacionado	desconocido n/-2/-1/0/+1/+2	importante/ irrelevante	trasero/ abajo a la derecha/ trasera izquierda/ frente/ frente derecho/ delantero izquierdo/ lado	normal/ desenfocar/ destello	marta, mio, máximo, maximo

Tabla 3: Propiedades peatonales

Tipo de objeto	Oclusión	Cabeza oclusión	Pies oclusión	Dirección	Movimienot	Altura	Pies oclusión	Extraño Pose	Encendiendo	Tamaño
peatonal/ mediocre	ninguno/ parcial/ lleno	verdadero/ FALSO	verdadero/ FALSO	NN/NE/ NW/SS/ SE/SO/ EE/WW	Moviente/ estacionario/ estacionado	adulto/ niño	verdadero/ FALSO	verdadero/ FALSO	normal/ desenfocar/ destello	marta, mio, máximo, maximo

Para crear la anotación para la detección y clasificación de vehículos, muchas empresas automotrices utilizan sus propias técnicas de detección de objetos y sus criterios de selección de propiedades. Algunas de las propiedades elegidas y el enfoque de detección de objetos utilizado se muestran en las tablas anteriores. La Tabla 1 clasifica el vehículo detectado como automóvil, autobús, camión, otro vehículo y sin descripción. El vehículo detectado se caracteriza además mediante el uso de diez propiedades, como oclusión, oclusión inferior, dirección, movimiento, asignación de carril, detección de cambio de carril, rotación, pose, iluminación y medición (tamaño) del cuadro delimitador. La Tabla 2 clasifica el vehículo de dos ruedas detectado como

II. Modelo propuesto

Los diversos cuadros de un video dado se alimentan como datos de entrada a nuestro modelo propuesto. Se espera que el modelo encuentre objetos tales como vehículos, vehículos de dos ruedas y peatones usando un algoritmo de detección de objetos y extraiga sus propiedades como oclusión, pose, dirección, etc. usando técnicas de clasificación de objetos. Se supone que el módulo detecta los carriles sobre la marcha utilizando el algoritmo de detección de carriles. Al rastrear todos los objetos usando un algoritmo apropiado, el modelo tiene que identificar los movimientos y registrar si cambian de carril. Estos pasos se muestran claramente en la fig. 1. a continuación El modelo se divide en cuatro sub

modelos Son detección de objetos, identificación de propiedades durante la clasificación, detección de carril y seguimiento de cada objeto detectado para detección de movimiento y cambio de carril. Los detalles del modelo propuesto se describen a continuación.

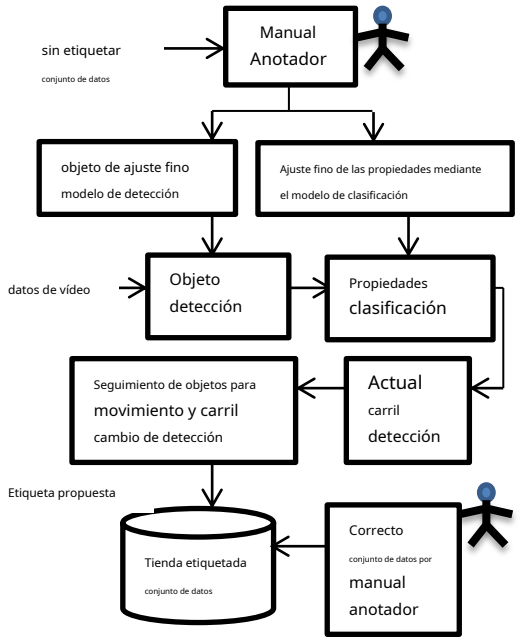

Fig. 1. Diagrama de bloques del modelo propuesto

A. Conjuntos de datos:

Utilizamos un conjunto de datos personalizado. Las muestras se recolectaron de los videos al costado de la carretera de los países asiáticos. Se instaló una cámara web normal en la parte superior del automóvil. Los caminos se clasificaron como carreteras y caminos de la ciudad. Algunas carreteras tenían la mediana/barrera central y algunas carreteras no la tenían. Los videos de entrada se anotaron manualmente como vehículos, vehículos de dos ruedas y peatones junto con sus características, como oclusión, oclusión inferior, oclusión de la cabeza, dirección, movimiento, pose, identificación de carril, cambio de carril, etc. El conjunto de datos personalizado recopilado manualmente tenía 28,450 objetos anotados a lo largo con las propiedades. Esto se hizo con 29 anotadores y tomó 4 días, cada día trabajaron durante aproximadamente 8 horas (55680 minutos). Por lo tanto, el tiempo promedio necesario para anotar cada objeto fue de aproximadamente 2 minutos.

B. Detección de objetos:

El primer paso de nuestra herramienta de anotación automática es la detección de objetos. En esta etapa, nuestra herramienta detecta los objetos como vehículos, vehículos de dos ruedas o peatones. Si el objeto se detecta como vehículo, la herramienta clasifica el objeto como automóvil/autobús/camión/otro vehículo/sin descripción. Si el objeto se detecta como un vehículo de dos ruedas, el sistema debe clasificarlo aún más como ciclomotor. /motociclista /ciclista /otro-vehículo de dos ruedas /anodino. El ser humano se identifica como anodino/peatón. Aquí, la propiedad no descriptiva se usa para las tres categorías. Determinamos si la altura y el ancho del cuadro delimitador detectado son inferiores a 30 píxeles. Si es así, asignamos la propiedad de ese objeto como no descriptivo. En algunos escenarios los vehículos o vehículos de dos ruedas vienen en dirección opuesta pero el camino está dividido por una mediana/barrera. Entonces el vehículo o vehículo de dos ruedas se clasifica como indescriptible. Se utilizan muchos algoritmos de detección de objetos en el aprendizaje profundo, como R-CNN, Fast-R-CNN, Faster-R-CNN, YOLO, SSD, Retinanet, etc. Para nuestro paso de detección de objetos, usamos YOLO[4] y Retinanet-50 [12] algoritmos. Los resultados obtenidos se comparan y muestran en nuestra sección de análisis de resultados. Figura 2a. a continuación se muestra el objeto detectado y la figura 2b muestra la segmentación de los objetos detectados. Los objetos segmentados luego se pasan al módulo de clasificación donde las propiedades de los objetos se identifican automáticamente además de clasificarlos más.

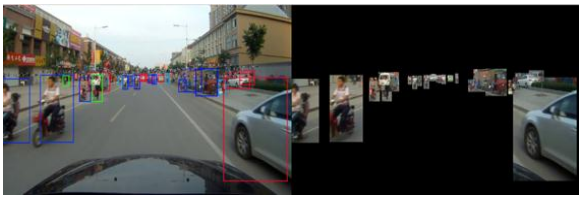

Fig. 2: (a) Detección de objetos (b) segmentación del objeto detectado

C. Clasificación y asignación de propiedades:

Los resultados obtenidos del módulo de detección de objetos se trasladan al módulo de asignación y clasificación de propiedades. Las Tablas 1 a 3 explicadas anteriormente describen las diversas propiedades que se identificarán automáticamente para cada objeto. Si el objeto es un vehículo, las propiedades a identificar son: tipo de objeto, oclusión, oclusión inferior, dirección, movimiento, asignación de carril, detección de cambio de carril, rotación, pose, iluminación y tamaño. Si el objeto detectado es un vehículo de dos ruedas, las propiedades que se asignarán son: oclusión del tipo de objeto, oclusión de la cabeza, oclusión de los pies, dirección, movimiento, asignación de carril, rotación, pose, iluminación y tamaño. Si el objeto detectado es un peatón entonces las propiedades a asignar son: tipo de objeto oclusión, oclusión de la cabeza, oclusión de los pies, dirección, movimiento, altura, pose extraña, iluminación y tamaño. Aquí, el tipo de objeto y el tamaño de todos los objetos se han detectado mediante el algoritmo de detección de objetos. La asignación de carril, la detección de cambio de carril y el movimiento se realizan mediante el algoritmo de seguimiento y detección de carril como módulos tercero y cuarto. Las propiedades restantes para todos los objetos son: oclusión, oclusión del fondo, oclusión de la cabeza, oclusión de los pies, dirección, rotación, altura,

pose, pose extraña e iluminación. Todos ellos se detectan mediante el algoritmo de clasificación. Aquí la oclusión significa cuánto está ocluido el objeto. Puede ser parcial, total o ninguno. La oclusión inferior significa que la parte inferior del vehículo no es visible. La oclusión de la cabeza y los pies se refiere a la cantidad de oclusión que han encontrado la cabeza y los pies del conductor de dos ruedas/peatón. La dirección se refiere a la dirección del vehículo/vehículo de dos ruedas. Usando métodos de clasificación podemos decir si el objeto se aproxima, precede o se encuentra al lado de nuestro vehículo del ego. Pero para el peatón, se clasifica como una de las ocho direcciones. La rotación depende de la dirección. Si el vehículo se aproxima o precede, entonces la rotación es relevante; de lo contrario, es irrelevante. La altura para un peatón es clasificar si el peatón es un adulto o un niño. La pose para el vehículo/vehículo de dos ruedas es clasificarlo según el lado visible. Eso es delante o detrás, etc. Pero la pose extraña para un peatón se refiere a casos en los que puede estar cargando a un bebé o haciendo algo en la carretera. La propiedad final se trata de la iluminación en la que clasificamos la claridad del objeto. Puede ser claro, deslumbrante o poco nítido. La siguiente figura 3 muestra las posibles direcciones de un peatón contra el ego car. Muestra ocho direcciones de un peatón y un peatón moviéndose a lo largo de la dirección Noroeste (NW). Puede ser claro, deslumbrante o poco nítido. La siguiente figura 3 muestra las posibles direcciones de un peatón contra el ego car. Muestra ocho direcciones de un peatón y un peatón moviéndose a lo largo de la dirección Noroeste (NW). Puede ser claro, deslumbrante o poco nítido. La siguiente figura 3 muestra las posibles direcciones de un peatón contra el ego car. Muestra ocho direcciones de un peatón y un peatón moviéndose a lo largo de la dirección Noroeste (NW).

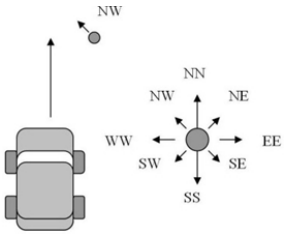

Fig. 3: Dirección del peatón

Para clasificar las propiedades de cada objeto, hemos utilizado dos algoritmos de clasificación, a saber, VGG-19[13] y Resnet-50[14]. Los resultados se muestran en la sección de análisis de resultados. Las siguientes figuras 4(a) a 4(f) muestran las propiedades de varias categorías de objetos.

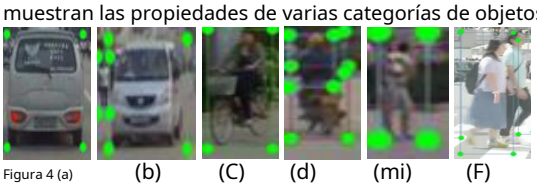

Figura 4 (a) (b) (C) (d) (mi) (F)

La figura 4a muestra el tipo de objeto como automóvil y sus propiedades son: oclusión: ninguna, oclusión inferior: falsa, dirección: anterior, rotación: relevante, pose: atrás y relámpago: normal. La figura 4b muestra el tipo de objeto como automóvil y sus propiedades son oclusión: ninguna, oclusión inferior: falsa, dirección: que se aproxima, rotación: relevante, pose: frontal izquierda e iluminación: normal. La figura 4c muestra el tipo de objeto como ciclista, y sus propiedades son: oclusión: ninguna, oclusión cabeza: falso, oclusión pies: falso, dirección: acercándose, rotación: relevante, pose: frontal izquierda e iluminación: normal. La figura 4d tiene un tipo de objeto: motociclista, y sus propiedades son: oclusión: ninguna, oclusión de la cabeza: falsa, oclusión de los pies: falsa, dirección: anterior, rotación: relevante, pose: atrás a la izquierda e iluminación: normal. En la figura 4e el tipo de objeto es peatón, y sus propiedades son: oclusión: ninguna, oclusión cabeza: falsa, oclusión pies: falsa, dirección: WW, altura: adulto, pose extraña: verdadera e iluminación: normal. En la figura 4f el tipo de objeto es peatón, y sus propiedades son: oclusión: parcial, oclusión cabeza: falsa, oclusión pies: verdadera, dirección: WW, altura: adulto, pose extraña: falsa e iluminación: normal. Nuestro próximo módulo es la detección de carril.

D. Detección de carril

En nuestro módulo de detección de carriles hemos considerado hasta 6 carriles. Están etiquetados como desconocidos, -2, -1, 0, 1, 2. Aquí 0 se refiere al carril en el que se mueve nuestro vehículo ego, -1 se refiere al carril en el lado izquierdo inmediato, -2 se refiere a dos carriles (más lejano) en el lado izquierdo, 1 se refiere al carril en el lado derecho inmediato de nuestro vehículo ego y 2 se refiere a dos carriles (más lejano) en el lado derecho del vehículo ego. Pero desconocido se refiere al objeto objetivo que no está en el camino del vehículo del ego. Por ejemplo, el objetivo puede estar estacionado en el área de estacionamiento o un peatón está parado en el sendero para peatones o el vehículo objetivo se está moviendo en el lado opuesto de la carretera donde la carretera está dividida por una mediana/barrera. Hemos elegido el algoritmo de detección de carriles para vehículos autónomos de Udacity[15] y el algoritmo de aprendizaje profundo de LaneNet[16]. Los resultados se comparan y se dan en la sección de análisis de resultados. La salida de LaneNet se muestra en la figura 5a, en la que se muestra la instancia de segmentación del carril. Las cuatro marcas de carril detectadas en la carretera están coloreadas con líneas de color rosa, azul, verde y amarillo. Aquí algunas líneas no estaban correctamente segmentadas. Para superar este problema, se usó la transformada de Hough para dibujar la línea recta sobre el carril segmentado y se muestra claramente en la figura 5b.

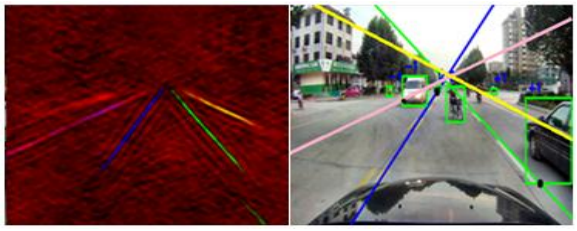

Fig. 5: (a) Segmentación de carril de instancia (b) Carril detectado con asignación de carril

Después de la detección de carril, asumimos la tarea de asignación de carril. Al carril en el que se mueve nuestro vehículo del ego se le asigna el número de carril 0. En la Figura 5b, se encuentra entre las líneas azul y verde etiquetadas como carriles. El carril que se encuentra entre las líneas rosa y azul es

asignado el número de carril -1. Está en el lado izquierdo del vehículo del ego. El carril en el extremo izquierdo de la línea rosa tiene asignado el número de carril -2. Al carril que se encuentra entre verde y amarillo se le asigna un número de carril + 1. Está en el lado derecho de nuestro vehículo del ego. El carril en el lado más a la derecha del vehículo ego se le asigna el número de carril +2. Cualquier objeto que no pertenezca a ningún carril recibe la asignación de carril desconocido. Aquí el problema es que si el vehículo precedente se mueve en la línea verde, ¿cómo podemos asignar si el vehículo se mueve en el carril número 0 o en el +1? Para este problema, uno puede tomar la línea inferior del objeto detectado y hacer una intersección con la línea de color subyacente etiquetada como carril. Según el punto de intersección, se puede encontrar si la parte máxima de la línea inferior se encuentra en el lado izquierdo o derecho del carril y, en consecuencia, se asigna el número de carril. Por ejemplo, en la figura 5b anterior, la línea inferior del automóvil detectado que precede cerca del vehículo ego se cruza con el carril de color verde y su parte máxima cae en el carril +1. Entonces, la asignación de carril para este automóvil se elige como número de carril +1.

E Seguimiento

Este es el último paso de nuestra herramienta de anotación completamente automática. En el primer paso, se detectaron vehículos, vehículos de dos ruedas y peatones mediante el uso de un algoritmo de detección de objetos. Los objetos detectados se pasaron a nuestro algoritmo de clasificación para averiguar las propiedades de cada objeto, como la oclusión parcial de vehículos o peatones que se mueven en dirección al norte (NN) y que llevan a un bebé en una posición extraña, etc. El tercer paso fue detectar el carril por el que circula el vehículo. Luego, los resultados se pasan a este paso de seguimiento para identificar si los vehículos o vehículos de dos ruedas detectados están en movimiento, estacionados o inmóviles. Usando la selección de propiedad de movimiento, se puede encontrar si el vehículo o vehículo de dos ruedas detectado está cambiando su carril usando la propiedad de identificación de cambio de carril. Se crea una identificación única para cada objeto en cada marco. Para el seguimiento, elegimos el algoritmo de detección y seguimiento de vehículos autónomos de Udacity [17] y el algoritmo de aprendizaje profundo de seguimiento de objetos múltiples Deep SORT [18]. Los resultados se informan en la sección de análisis de resultados. En el algoritmo de seguimiento, cada objeto detectado recibe una identificación de seguimiento. Esta ID se utiliza como una ID única del objeto y se almacena junto con las propiedades de los objetos. La última propiedad que necesitamos encontrar es el movimiento. El objetivo aquí es encontrar si el objeto está en movimiento, estacionado o estacionario. Para esta detección de movimiento, las entradas son cuadros delimitadores de objetos con ID de seguimiento y número de carril asignado. El cuadro delimitador del objeto detectado en el cuadro anterior y el cuadro delimitador del objeto detectado en el cuadro actual se marcan para ver si poseen la misma identificación de seguimiento. Si es así, se pasa a la técnica de extracción de características ORB (Oriented FAST and Rotated BRIEF) [19] que proporciona puntos clave de la imagen del objeto detectado. Los puntos clave de una imagen son características únicas extraídas de la imagen. Los puntos clave obtenidos del marco anterior y actual para cada objeto se pasan al algoritmo BF Matcher (Brute Force Matching)[19] para encontrar el punto de coincidencia de ambas imágenes de objetos. A partir de los puntos coincidentes, se puede obtener la distancia entre las dos coordenadas de la imagen y se denomina distancia en píxeles de dos ROI (Región de interés). Uno puede recopilar la distancia de todos los puntos coincidentes y encontrar su distancia media. Si la distancia media es mayor que la distancia de seis píxeles, ese objeto se asigna a la clase de movimiento. Si la distancia de píxeles es inferior a seis píxeles, el objeto todavía está en cualquiera de los carriles (-2,-1,0,1,2). Entonces el objeto pertenece a la clase estacionaria. Si la distancia de píxeles es inferior a seis píxeles pero el objeto está en el carril desconocido, el objeto se asigna a la clase de estacionamiento. La figura 6 a continuación muestra la detección de movimiento del objeto detectado.

Fig. 6: Seguimiento de objetos y detección de propiedades de movimiento

III Análisis de resultados:

En esta sección analizamos los resultados obtenidos utilizando nuestra herramienta de anotación totalmente automática. Aquí elegimos el mejor método para nuestro algoritmo final.

A. Detección de objetos:

Para la detección de objetos, las métricas utilizadas para el estudio comparativo entre YOLO y Retinanet son la pérdida de entrenamiento, la precisión media (AP) y el AP medio (mAP)[20]. Para la prueba se utilizó un video con 530 fotogramas cada uno de resolución 1035 x 1800 píxeles. En promedio, cada cuadro tenía alrededor de 15 objetos. Anotamos manualmente (7950 objetos) y los utilizamos para este experimento. La Tabla 4 a continuación muestra la precisión promedio (AP) de varios objetos y la compara con los algoritmos YOLO y RetinaNet-50. El mAP de YOLO es 34,35, mientras que el mAP de Retinanet-50 es 48,3. La Tabla 5 a continuación muestra la Precisión (%) de varios objetos detectados y los compara con los algoritmos YOLO y RetinaNet-50. De la tabla 5, la precisión media general (9 clases de objetos) de YOLO es 60.12% y eso

de Retinanet-50 es del 82,13%. La figura 9a muestra la pérdida de entrenamiento de YOLO y la figura 9b muestra la pérdida de entrenamiento de Retinanet-50. El Retinanet-50 tiene una pérdida menor que YOLO. Por lo tanto, elegimos el algoritmo Retinanet-50 como nuestro algoritmo de detección de objetos.

B. Clasificación de propiedades

A partir de la salida de nuestro algoritmo de detección de objetos, a saber, Retinanet-50, los objetos detectados se pasan al algoritmo de clasificación de propiedades. Aquí notamos la falta de precisión en el algoritmo de detección de objetos. La Figura 7 muestra un objeto en el que la oclusión se clasifica como ninguna durante la anotación manual, pero el algoritmo de clasificación automática lo ha declarado como oclusión parcial debido a que el algoritmo de detección de objetos no ha dibujado correctamente el cuadro delimitador. La Tabla 6 muestra el estudio comparativo entre VGG-19 y Resnet-50 para las propiedades de oclusión de vehículos y dirección de peatones. Las figuras 9c y 9e muestran la pérdida de entrenamiento y la pérdida de validación de VGG-19 y Resnet-50 para las propiedades de oclusión de vehículos y dirección de peatones. Resnet-50 tiene una pérdida menor tanto para el entrenamiento como para la validación. Las figuras 9d y 9f muestran el promedio de entrenamiento y el promedio de validación de VGG-19 y Resnet-50 para las propiedades de oclusión de vehículos y dirección de peatones. El Resnet-50 tiene el promedio más alto durante la fase de entrenamiento y validación. El Resnet-50 ha brindado una alta precisión (en general, 97%) que el modelo VGG-19. Por lo tanto, se selecciona el algoritmo ResNet-50 para la clasificación de propiedades.

Fig. 7: clasificación incorrecta debido a un tamaño de cuadro delimitador incorrecto

C. Detección de carril

El algoritmo de detección de carriles de Udacity no pudo detectar los carriles para los conjuntos de datos de carreteras asiáticas y se muestra en la figura 8a a continuación. Detecta los objetos pero no pudo detectar correctamente el carril. Por lo tanto, se elige LaneNet para la técnica de detección de carriles. El número de objetos detectados en el carril correcto fue 6678. Nuestro algoritmo de detección de objetos propuesto no detecta algunos de los objetos y se muestran en la figura 8b como círculos rojos (peatones y ciclomotores). La cantidad de objetos anotados manualmente para el video que tiene 530 cuadros es 7950. Por lo tanto, el algoritmo LaneNet produjo una precisión del 84 % en comparación con la anotación manual.

D. Seguimiento de objetos

Las métricas utilizadas para el seguimiento de objetos múltiples son Precisión de seguimiento de objetos múltiples (MOTA)[21], Precisión de seguimiento de objetos múltiples (MOTP)[21], Mayormente rastreados (MT)[21] y Mayormente perdidos (ML)[21]. La comparación entre el algoritmo de seguimiento de Udacity y el algoritmo Deep SORT utilizando las cuatro métricas anteriores se muestra en la Tabla 7.

Tabla 4: Precisión promedio de YOLO-V2 frente a RetinaNet-50

	Auto (AP)	Autobús (AP)	Camión (AP)	Otro-Vehículo (AP)	motociclista (AP)	Motorcyclist (AP)	Ciclista (AP)	Otros dos-Rueda (AP)	Peatonal (AP)
yolo	38.6	35.2	40,0	30,9	34.6	33.3	31.1	29.8	35.7
Retinanet-50	54.1	56.7	48.7	41.6	49.8	41.3	50.2	42.7	49.6

Tabla 5: Precisión de YOLO-V2 frente a RetinaNet-50

	Auto (%)	Autobús (%)	Camión (%)	Otro-Vehículo (%)	motociclista (%)	Motor-ciclista (%)	Ciclista (%)	Otros dos-Wheeler (%)	Peatonal (%)
yolo	74	60	80	40	57.3	60.8	40	75	54
Retinanet-50	86,9	87	93	81	82,9	71	80	69	88.4

Tabla 6: Precisión de VGG-19 frente a ResNet-50

	oclusión del vehículo	dirección peatonal
VGG-19	86.6	79.5
ResNet-50	97.45	96.8

Tabla 7: Métricas para seguimiento (proyecto Udacity frente a Deep SORT)

	MOTA	MOTP	MONTE	ML
Seguimiento de Udacity	15.9	69.8	6,4%	47,9%
CLASIFICACIÓN profunda	71.4	79.1	34%	18,2%

Tabla 8: Tiempo de procesamiento en CPU vs GPU

	sistema de CPU	sistema GPU
Detección de objetos	13,2 minutos	51 segundos
Clasificación de objetos	12,5 minutos	44 segundos
Identificación de carril	7,5 minutos	22 segundos
seguimiento de objetos	10,5 minutos	37 segundos

Fig. 8 (a) Detección de carril Udacity (b) Detección de carril LaneNet

La precisión del seguimiento de objetos múltiples (MOTA) con SORT profundo es del 71,4 %. Por lo tanto, Deep SORT se elige como el algoritmo de seguimiento de objetos para el desarrollo de nuestra herramienta de anotación totalmente automática.

E. Tiempo de procesamiento:

El tiempo necesario para completar los diversos módulos para el video de prueba (530 fotogramas) se muestra en la Tabla 8 anterior. Hemos probado nuestro video utilizando el sistema basado en CPU cuya especificación es el procesador Intel i5, 16 GB de RAM y el sistema operativo Ubuntu 16.04 y la especificación de nuestro sistema basado en GPU es el procesador Intel XEON, la tarjeta gráfica NVIDIA Quadro P1000, 32 GB de RAM y Ubuntu 16.04 Sistema operativo. El video de entrada con 530 fotogramas tarda 43 minutos en completar el proceso de anotación automática en el sistema de la CPU, mientras que el sistema de la GPU ha tardado solo 2,58 minutos. La anotación manual por un solo anotador ha tomado 3060 minutos. Por lo tanto, afirmamos que nuestra herramienta de anotación automática propuesta es razonablemente rápida (alrededor de 1200 veces en comparación con un sistema de GPU) y también precisa.

IV.conclusión:

Hemos descrito brevemente algunos algoritmos que se utilizan para la detección de objetos, clasificación, identificación de carriles y modelos de seguimiento de objetos. Hemos analizado los algoritmos y elegido el mejor algoritmo en cada modelo. Los algoritmos elegidos se utilizaron en nuestra herramienta de anotación totalmente automática para crear el conjunto de datos de entrenamiento automático para el automóvil autónomo. Nuestros resultados (basados en el video de prueba con 530 fotogramas, cada uno de los cuales contiene unos 15 objetos) muestran que Retinanet-50 es el mejor algoritmo para el modelo de detección de objetos, ya que tiene una precisión del 82 %. El ResNet-50 es elegido para la clasificación de propiedades

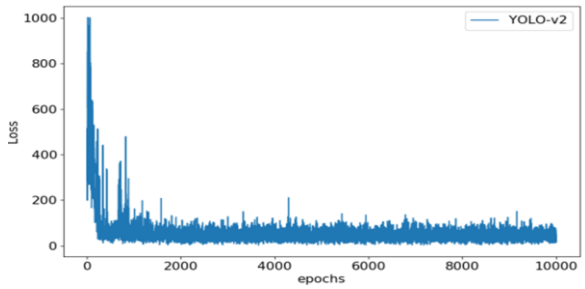

Fig. 9. (a) Pérdida de entrenamiento de YOLO-V2

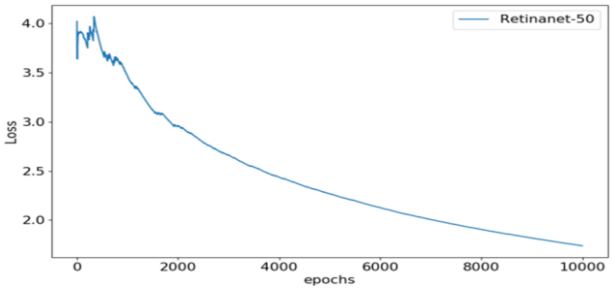

Fig. 9. (b) Pérdida de entrenamiento de RetinaNet-50

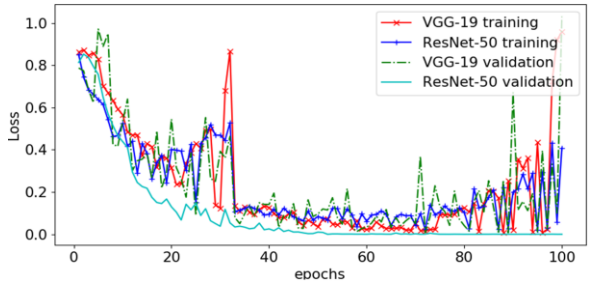

Fig. 9. (c) Pérdida de oclusión del vehículo

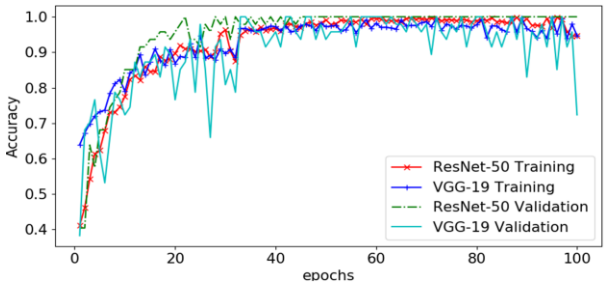

Fig. 9. (d) Precisión de la oclusión del vehículo

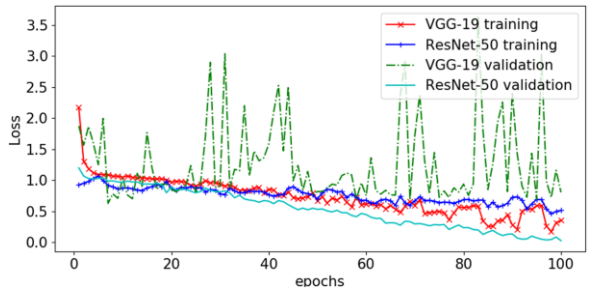

Fig. 9. (e) Pérdida de dirección del peatón

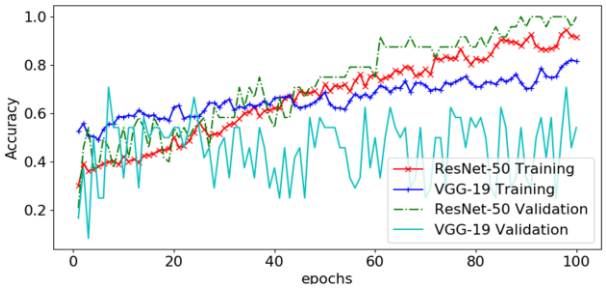

Fig. 9. (f) Precisión de la dirección del peatón

algoritmo que proporciona un 97% de precisión. El LaneNet es elegido como el mejor algoritmo para la identificación de carriles cuya precisión es del 84%. Para el seguimiento de objetos, el algoritmo Deep SORT se elige como el mejor algoritmo con un 71 % de precisión. La precisión promedio de nuestra herramienta de anotación completamente automática es del 83 %, que es un 17 % menor que la anotación manual. Nuestra herramienta de anotación totalmente automática admite la anotación manual para las correcciones. Y el tiempo total de procesamiento usando el sistema de CPU es de 43 minutos y 2,58 minutos en el sistema de GPU. La anotación manual tomó alrededor de 3060 minutos para un anotador. Por lo tanto, nuestra herramienta de anotación completamente automática que utiliza el sistema GPU es unas 1200 veces más rápida que el anotador manual.

RECONOCIMIENTO:
El trabajo propuesto se llevó a cabo en el TIFAC-CORE en el Centro de Investigación de Infotrónica Automotriz en VIT, Vellore. Nos gustaría agradecer a DST, Gobierno de la India por el apoyo brindado al centro.

Referencia:

1. Informe sobre el estado mundial de la seguridad vial de la OMS de 2015, Organización Mundial de la Salud (2015), http://www.who.int/violence_injury_prevention/road_safety_status/2015/en/, consultado el 11 de febrero de 2019
2. M. Kyriakidis, R. Happee, JC de Winter, 'Opinión pública sobre conducción automatizada: resultados de un cuestionario internacional entre 5000 encuestados', Transporte. Res. Parte. F: Psicología del Tránsito. Behav., 32 (2015), págs. 127-140, 10.1016/j.trf.2015.04.014
3. Declaración preliminar de política de la NHTSA sobre vehículos automatizados Administración Nacional de Seguridad del Tráfico en las Carreteras, Washington, DC (2013), págs. 1-14
4. B. Russell, A. Torralba, K. Murphy, WT Freeman. 'LabelMe: una base de datos y una herramienta basada en la web para la anotación de imágenes', International Journal of Computer Vision, Springer, 2007.
5. J. Redmon, S. Divvala, R. Girshick y A. Farhadi, 'Solo miras una vez: detección unificada de objetos en tiempo real', Actas de la conferencia IEEE sobre visión artificial y reconocimiento de patrones, 2016, págs. 779– 788
6. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein,A. Berg y L. Fei-Fei, 'Desafío de reconocimiento visual a gran escala de Imagenet', IJCV, 2015.
7. H. Su, J. Deng y L. Fei-Fei, 'Crowdsourcing anotaciones para la detección visual de objetos', Taller de Computación Humana de la AAAI, 2012.
8. Dim P. Papadopoulos, Jasper RR Uijlings, Frank Keller, Vittorio Ferrari, 'Entrenamiento de detectores de clases de objetos con supervisión de clics', arXiv:1704.06189, 2017
9. Adithya Subramanian, Anbumani Subramanian, 'Anotación de un clic con guía jerárquica Objeto Detección', arXiv:1810.00609v1, 2018
10. Dim P. Papadopoulos Jasper RR Uijlings Frank Keller Vittorio Ferrari, 'No necesitamos cuadros delimitadores: entrenamiento de detectores de clases de objetos usando solo verificación humana', arXiv: 1602.08405v3, 2017
11. Zhujun Xiao, Yanzi Zhu, Yuxin Chen, Ben Y. Zhao, Junchen Jiang, Haitao Zheng, 'Abordar el sesgo de entrenamiento a través de la anotación de imagen automatizada', arXiv:1809.10242, 2018
12. T.-Y. Lin, P. Goyal, R. Girshick, K. He y P. Dollar. 'Pérdida focal para la detección de objetos densos', transacciones IEEE sobre análisis de patrones e inteligencia artificial, 2018.
13. Karen Simonyan y Andrew Zisserman, 'Redes convolucionales muy profundas para el reconocimiento de imágenes a gran escala', arXiv:1409.1556, 2014
14. K. He, X. Zhang, S. Ren y J. Sun. 'Aprendizaje residual profundo para el reconocimiento de imágenes', conferencia IEEE sobre visión artificial y reconocimiento de patrones (CVPR), 2016.
15. Proyecto de nanogrados de vehículos autónomos de Udacity, "Encontrar carril líneas", https://in.udacity.com/nanodegree, consultado el 21 de febrero de 2019
16. Ze Wang, Weiqiang Ren, Qiang Qiu, 'LaneNet: redes de detección de carriles en tiempo real para conducción autónoma', arXiv:1807.01726, 2018
17. Proyecto Udacity Self-Driving Car Nanodegree, "Detección y seguimiento de vehículos", https://in.udacity.com/nanodegree, consultado el 13 de marzo de 2019
18. N. Wojke, A. Bewley, D. Paulus, 'Seguimiento simple en línea y en tiempo real con una métrica de asociación profunda' ICIP, pp. 3645-3649, 2017.
19. S. Jayaraman, S. Esakkirajan, T. Veerakumar, 'Digital Image Processing', publicación de Tata McGraw Hill, edición india (2015).
20. Liu.I, Ozsum.T, 'Mean Average Precision', Enciclopedia de sistemas de bases de datos. Springer, 2009.
21. Anton Milan, Laura Leal-Taixe, Ian Reid, Stefan Roth, Konrad Schindler, 'MOT16: Un punto de referencia para el seguimiento de múltiples objetos', arXiv:1603.00831, 2016