

ZebrAI Crossing: Detecció de passos de zebra i semàfors de vianants

Albert Capdevila Estadella (1587933), Levon Kesoyan Galstyan (1668018), Luis Martínez Zamora (1668180)

Abstract — **ZebrAI Crossing** és un sistema de visió per computador que detecta passos de zebra i semàfors de vianants, i en determina l'orientació, la posició i l'estat. Per fer-ho, es combinen tècniques tradicionals com filtratge de soroll, binarització adaptativa, detecció de vores, transformades de Hough i RANSAC per extreure els atributs del pas de zebra amb tècniques de deep learning com YOLO per a la detecció de semàfors. El model s'ha avaluat sobre 7000 imatges, mostrant resultats que, tot i no ser suficientment bons, reflecteixen la complexitat del problema.

Keywords — Zebra crossing detection, edge detection, Hough transform, RANSAC, image binarization, YOLO.

Repository —  [albert-ce/ZebrAI-Crossing](https://github.com/albert-ce/ZebrAI-Crossing)

1 INTRODUCCIÓ

ZebrAI Crossing és un model de visió per computador que, donada una imatge en un entorn urbà, detecta si hi apareix un pas de zebra i en determina la seva orientació i posició respecte la càmera. A més a més, també detecta la presència de semàfors per a vianants i indica el seu estat (en vermell, en verd...).

1.1 Objectius

- Identificar si apareix un pas de zebra a la imatge.
En cas afirmatiu:
 - Determinar l'orientació del pas de zebra.
 - Determinar la posició de l'inici del pas de zebra.
- Identificar si apareix un semàfor de vianants a la imatge.
En cas afirmatiu:
 - Indicar si el semàfor es troba en verd.

2 ESTAT DE L'ART

Aquests són els recursos que ens han resultat més útils de tots els que hem investigat (Veure Referències):

2.1 Projectes

ImVisible [2]

El projecte tracta del desenvolupament d'una aplicació mòbil que detecta passos de zebra, la seva orientació i, si hi ha semàfor, identifica si està en verd o vermell. Tot el procés de reconeixement d'imatges es fa mitjançant xarxes neuronals convolucionals. L'aspecte que ens ha sigut més útil és el **dataset** que utilitza, anomenat Pedestrian Traffic Lights, perquè té moltes imatges anotades amb atributs que ens interessin.

2.2 Articles

Zebra-crossing Detection for the Partially Sighted [7]

Aquest article utilitza transformacions de Hough i detecció de variacions d'intensitat per a identificar les línies defineixen el pas de zebra. A més, es proven diferents mètodes per diferenciar-los d'imatges d'escales, un basat en homografies i els altres amb restriccions basades en els punts de fuga.

Zebra-crossing detection based on cascaded Hough transform principle and vanishing point characteristics [9]

L'article presenta un estudi per detectar passos de zebra utilitzant un filtratge de Gauss, detecció de vores de Canny i una transformada en cascada de Hough basada en el punt de fuga, sense extreure regions d'interès. També es compara amb la transformada de Hough estàndard, demostrant que la tècnica en cascada obté millors resultats.

ZebraRecognizer: Pedestrian crossing recognition for people with visual impairment or blindness [11]

Aquest article presenta una app per invidents que detecta passos de zebra en temps real mitjançant una homografia per corregir la perspectiva, sobre la qual aplica una detecció de vores. Finalment, utilitza regressió ortogonal per identificar una línia comuna entre els segments detectats.

3 PROPOSTA

3.1 Descripció de les dades

En aquest projecte, hem utilitzat un conjunt de dades fabricat a partir de dos datasets diferents: **Pedestrian Traffic Lights** [2] i **GlobalStreetscapes** [16]. Aquest conjunt de dades s'ha utilitzat per avaluar i provar la detecció de passos de zebra i els seus atributs, així com per provar la detecció de l'estat dels semàfors per a vianants.

Pedestrian Traffic Lights proporciona una gran quantitat d'imatges anotades amb l'estat del semàfor de vianants i dos punts que defineixen un segment perpendicular al pas de zebra. Així doncs, hem extret la orientació del pas de zebra i la posició de l'inici d'aquest amb els càlculs:

$$\theta = \tan^{-1} \left(\frac{x_2 - x_1}{y_2 - y_1} \right) \quad (x, y) = (x_i, y_i : y_i > y_{3-i})$$

A més, per a fer una bona avaluació, ha sigut necessari afegir imatges que no continguin passos de zebra, motiu pel qual hem decidit incorporar imatges de **GlobalStreetscapes**.

Per tant, els atributs finals que té cada imatge del dataset que hem construït són: **[file, zebra, mode, blocked, x, y, theta_rad, theta_deg]**, on **zebra** indica si hi ha pas de zebra o no, **mode** indica l'estat del semàfor, **blocked** indica si el semàfor no és visible (bloquejat), **(x, y)** és la posició de l'inici del pas de vianants i **theta** és l'angle entre la línia de creuament i l'eix vertical de la imatge (eix y). És a dir, la càmera hauria de girar θ° per poder creuar el pas de zebra avançant cap endavant.¹

¹És una simplificació que no té en compte la distorsió provocada pels punts de fuga. Les rectes paral·leles a la realitat no sempre ho són a la imatge, pel que el gir desitjat no és exactament θ° .

3.2 Tècniques a utilitzar

Aquest projecte combina dues metodologies: **tècniques tradicionals** per a detectar els passos de zebra i els seus atributs, com ara morfologia matemàtica, detecció de vores,... i **tècniques basades en deep learning** per a la detecció dels semàfors i el seu estat, amb eines com ara YOLO [1].

4 EXPERIMENTS, RESULTATS I ANÀLISI

4.1 Detecció de semàfors i ROI amb YOLO

En la primera part del projecte, hem utilitzat un model preentrenat de YOLO que detecta passos de zebra i semàfors. Respecte als passos de zebra, hem obtingut la ROI que hem fet servir per al tractament posterior, i respecte als semàfors, el model ja ens dona informació sobre si aquest es troba en vermell o en verd, assolint el nostre objectiu principal. A l'última secció d'aquest apartat es pot veure la avaluació que hem fet d'aquesta detecció.

4.2 Extracció dels atributs del pas de zebra

En aquest apartat hem usat la ROI prèvia per determinar l'orientació i la posició inicial del pas de zebra. Ho hem fet identificant-ne primer les vores paral·leles.

Les imatges amb les que hem treballat tenien molt **soroll impulsional**, i a més, normalment la pintura dels passos de zebra presenta molts desperfectes, pel que hem hagut d'aplicar un filtratge de soroll. N'hem provat quatre: mediana, Gaussià, bilateral (similar al Gaussià però mantenint millor les vores) i morfologia de màxims (dilate, per omplir forats a la pintura).

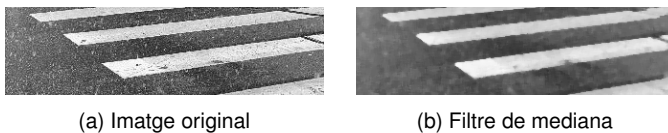


Fig. 1: Exemple del filtratge de soroll

El **filtre de mediana** ha donat un bon resultat visual: les barres del pas de zebra es veuen nítides i, en casos com el de la Figura. 1, on plou, elimina tot el soroll de la pluja, a diferència de les altres opcions.

Una vegada hem tingut la imatge filtrada, hem decidit treure'n una màscara binària per detectar-ne millor les vores.

El problema del top-hat:

En algunes imatges, les ombres han dificultat l'extracció de la màscara binària. La tècnica del **top-hat** és útil en situacions així, però ens hem trobat amb un problema: no hi ha un kernel que funcioni bé ni per a totes les imatges ni dins la mateixa imatge, ja que la mida i orientació de les barres varia amb la perspectiva. Un kernel petit només destaca les barres llunyanes, i un de gran no elimina les ombres. Així doncs, en aquests casos la precisió dependrà de les parts menys afectades per les ombres.

Totes les imatges són diferents, motiu pel qual aplicar un llindar constant a totes elles no era una opció viable. Hem utilitzat el llindar adaptatiu **Otsu**, implementat en la funció d'OpenCV. Ara bé, analitzant l'histograma d'algunes de les imatges, hem vist que l'Otsu no segueix la nostra intuïció d'un millor llindar: situar-lo a la vall entre els pics dels clars i els foscos.

Així doncs, hem trobat una solució per corregir el llindar que consisteix a fer una **mixtura de gaussianes** sobre mostres reduïdes

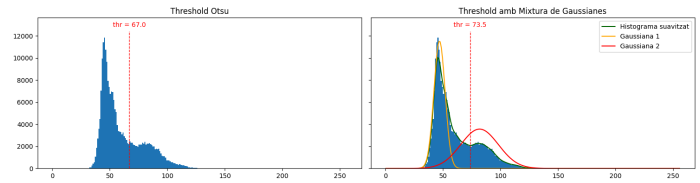


Fig. 2: Exemple d'histograma amb els llindars tractats

de la imatge (per eficiència) i escollir el mínim de l'histograma suavitzat entre els pics de les gaussianes estimades. A la Figura 2 es poden veure les diferències entre els dos mètodes.

Feta la binarització amb l'últim llindar, l'hem utilitzada per extreure les vores de les barres del pas de zebra. Hem provat dues tècniques: **Canny** i **LoG**. Totes dues han donat resultats similars, però hem continuat amb **LoG** per dues raons: no depèn d'hiperparàmetres com els llindars de Canny, i no fa vores fines, fet que restringeix menys la detecció de línies amb Hough.

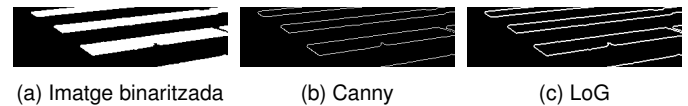


Fig. 3: Exemple de detecció de vores

Amb les vores del pas de zebra detectades, només ha faltat extreure les rectes amb **transformades de Hough**, però aquesta tècnica també depèn d'un llindar. Quan és molt alt no es detecten totes les línies, i quan és molt baix es detecten masses línies errònies. És una tècnica molt sensible a errors, com hem pogut comprovar quan la hem implementat a mà a mode de prova.² Per aconseguir millors resultats, hem fet aquest **mètode propi**:

1. Apliquem Hough amb llindar alt per obtenir poques línies fiables
2. Trobem les seves interseccions (punts de fuga)
3. Agrupem les interseccions properes amb clusterització
4. Apliquem Hough amb llindar baix per obtenir moltes línies
5. Eliminem les línies que no es dirigeixen als punts de fuga (errònies)

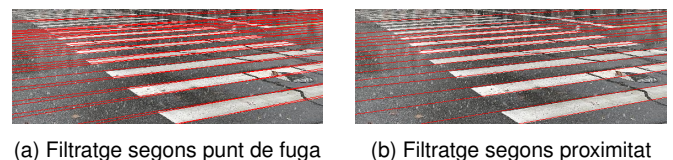


Fig. 4: Exemple de filtratge de línies

Aquest mètode recupera més línies, però introdueix duplicats. Els hem eliminat descartant les línies que intersequen dins la imatge amb angles més allunyats de la mitjana (Veure Fig. 4)

Tenint les línies filtrades, hem limitat la seva longitud segons les vores trobades anteriorment.

El següent pas ha estat detectar les línies verticals, que el mètode de Hough no detecta. Per fer-ho, hem utilitzat els extrems trobats de cada línia i, mitjançant RANSAC, hem seleccionat els dos punts que defineixen la recta amb més suport (és a dir, la més propera a més punts simultàniament). Un cop obtingudes aquestes dues línies verticals, intersequem les línies horitzontals amb elles per delimitar clarament el pas de zebra.

Per acabar, hem calculat l'angle del pas de zebra obtenint els punts mitjos entre les dues interseccions de cada línia horitzontal.

A partir d'aquests punts centrals, hem aplicat RANSAC per ajustar una recta i n'hem extret l'angle respecte a l'eix vertical. Per

²El notebook corresponent es pot veure al repositori.

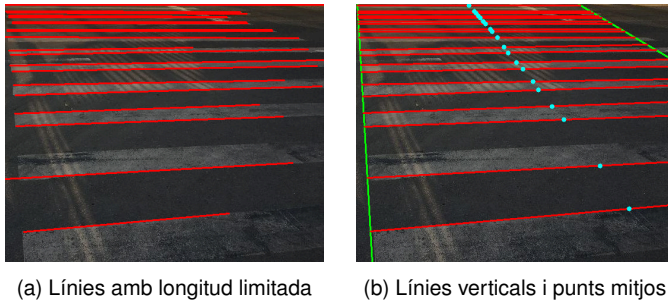


Fig. 5: Delimitació del pas de zebra reconstruïda

últim, també hem extret el punt mig més baix (més proper) com a punt d'inici del pas de zebra.

4.3 Resultats de rendiment

Tenint el model fet, només ha calgut fer una cerca d'hiperparàmetres i avaluar-ne els resultats. Hem dividit el conjunt de dades (unes **7000 imatges**) entre validació (25%) i prova (75%). La cerca aleatòria (Random Search) ha executat múltiples avaluacions amb combinacions d'hiperparàmetres, utilitzant com a mètrica el **MAE** (Mean Absolute Error), per la seva simplicitat i unitats interpretables. Les variables s'han ponderat així: x : 10%, y : 10%, θ_{rad} : 50% i θ_{deg} : 30%.

Els hiperparàmetres amb millor puntuació han sigut:

```
thr_option = otsu      ransac_niter = 1000
ransac_dist = 15       noise_option = median
med_k = 5              low_thr = 140
high_thr = 220         edges_option = canny
canny_thr1 = 25        canny_thr2 = 150
line_dist_thr = 50     centroids_sep = 120
```

Amb aquests, els resultats del conjunt de prova han sigut:

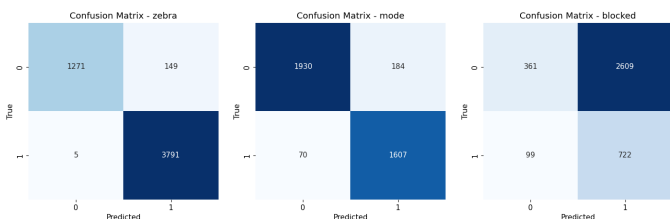


Fig. 6: Matrius de confusió amb el conjunt de test

Variable	x	y	θ_{rad}	θ_{deg}
MAE	268.6 px	318.6 px	0.5940 rad	34.04°

Taula 1: MAE amb el conjunt de test

Hem vist que les matrius de confusió són correctes exceptuant la de **blocked**. Aquestes depenen del model de YOLO, pel que no hem pogut fer gran cosa al respecte. Això sí, a nivell de riscos, és bo que els errors siguin **falsos positius** perquè d'aquesta manera no detectem semàfors quan aquests no existeixen.

Respecte als resultats de regressió, hem vist també que són força dolents. Hem prioritzat l'angle respecte la posició ja que és la part més extensa del projecte, i en aquest hem obtingut 34.04° de MAE. Aquests gran marge d'error es deu a la complicada detecció de les línies verticals, de les quals depèn molt el càlcul de l'angle. Igualment, estem contents amb els resultats perquè és un bon intent d'aplicar dins un cas real les tècniques vistes a classe.

5 CONCLUSIONS

Amb aquest projecte hem pogut integrar diferents tècniques de visió per computador per detectar passos de zebra i extreure'n els seus atributs, així com identificar l'estat dels semàfors per a vianants. Tot i això, durant el desenvolupament ens hem trobat amb dificultats importants.

En primer lloc, l'algorisme de **Hough** ha sigut difícil de controlar-ne els resultats. Depèn fortament dels paràmetres d'entrada, i petits canvis poden provocar que es detectin moltes línies errònies o que no se'n detecti cap. Això ha requerit diverses estratègies de filtratge per tal de millorar-lo.

A més, el fet que la **il·luminació fos tan diferent** entre imatges ha fet impossible poder extreure de manera fiable el pas de zebra. Ombres, reflexos i condicions ambientals com pluja han afectat molt negativament la binarització i la detecció de vores. Tot i aplicar mètodes adaptatius com Otsu o mixtura de gaussians, no hem pogut trobar un enfocament que sigui robust a totes les situacions.

Ens hem quedat amb ganes d'explorar més a fons tècniques com la detecció de **Haar-like features** i de trobar una solució més eficaç al problema del **top-hat**. Són aspectes que deixem oberts per a treballs futurs.

REFERÈNCIES

- [1] J. Redmon, *YOLO: Real-Time Object Detection*.
- [2] S. Yu et al., *ImVisible: Pedestrian Traffic Light Dataset, Lyt-Net Neural Network, and Mobile Application for the Visually Impaired*. GitHub repository.
- [3] N. Kuznetsov, *Crosswalks Detection using YOLO: A Supervised Method for Detecting Pedestrian Crosswalks*. GitHub repository.
- [4] CVC UAB, *Real-time detection of people and bicycles in pedestrian crossings*.
- [5] R. Santos, *PedestrianTrafficLightsAndCrosswalkDetection*. GitHub repository.
- [6] M. A. Rahman et al., *Zebra Crossing Detection and Time Scheduling Accuracy Enhancement Optimization Using Artificial Intelligence*. ResearchGate, 2021.
- [7] D. Bradley and A. Dunlop, *Zebra Crossing Detection for the Partially Sighted*.
- [8] D. Ahmetovic et al., *ZebraRecognizer: efficient and precise localization of pedestrian crossings*. 2014.
- [9] J. Zhang and L. Wang, *Zebra-crossing detection based on cascaded Hough transform principle and vanishing point characteristics*. De Gruyter, 2022.
- [10] P. Sharma and R. K. Singh, *Zebra Crossing Detection using Image Processing Techniques*. IJEESE, 2022.
- [11] Y. Chen et al., *ZebraRecognizer: Pedestrian crossing recognition for people with visual impairment or blindness*. Pattern Recognition, 2016.
- [12] Dkdkd. *july_6 Dataset*. Roboflow Universe, juny 2024.
- [13] Dkdkd. *capstone for detection Dataset*. Roboflow Universe, juny 2024.
- [14] Dkdkd. *capstone for detection1 Dataset*. Roboflow Universe, juny 2024.
- [15] Esera. *crosswalk Dataset*. Roboflow Universe, agost 2024.
- [16] UALSG. *Global Streetscapes Dataset for Crosswalk Detection*. GitHub repository.