XARXES NEURONALS I APRENENTATGE PROFUND

GRAU EN ENGINYERIA DE DADES

CURS 2024-2025

INFORME

BIRD'S EYE VIEW RECONSTRUCTION

Albert Guillaumet Mata, Lucia Garrido Rosas, Adrià Muro Gómez, David Morillo Massagué

Bellaterra, Juny de 2025

Introducció

En els darrers anys, la conducció autònoma ha avançat molt gràcies a sistemes que entenen i interpreten l'entorn del vehicle. Aquests sistemes processen gran quantitat d'informació visual per identificar obstacles, delimitar la calçada, reconèixer senyals i predir el comportament dels usuaris. Per això, és clau disposar de representacions espacials clares i útils per a la presa de decisions.

Al Centre de Visió per Computador (CVC) disposen d'un vehicle de conducció autònoma equipat **només amb càmeres frontals**. Per millorar la seguretat, és necessari ampliar la visió afegint càmeres a la part posterior i als laterals. Això permetria captar més informació de l'entorn, evitant punts morts. La generació d'una vista zenital o Bird's-Eye View (BEV) que integri totes aquestes imatges és una solució adequada per obtenir una visió global i estructurada.

La vista BEV ofereix una perspectiva des de dalt que facilita la comprensió de l'escena i és molt útil per a la planificació de trajectòries i detecció d'obstacles. Tot i això, obtenir aquesta vista a partir de múltiples càmeres és un repte, ja que els mètodes tradicionals basats en calibratge i geometria poden ser fràgils. En canvi, l'aprenentatge automàtic permet generar vistes BEV de manera més flexible i adaptativa.

Per això, aquest projecte investiga com **generar una imatge BEV a partir de les càmeres que envolten el vehicle**. Necessitem un model que entengui l'entorn a partir d'aquestes imatges i que sintetitzi la informació en una vista zenital clara i funcional. El sistema es desenvolupa en un entorn simulat per avaluar-ne la viabilitat i preparar aplicacions reals.

Hipòtesis

En el marc del projecte, s'han definit tres hipòtesis principals que orienten tant el disseny com l'avaluació del sistema proposat. Aquestes hipòtesis parteixen de supòsits plausibles dins del context de la percepció per visió artificial aplicada a la conducció autònoma, i pretenen validar la viabilitat del sistema en diferents escenaris i condicions.

- **Semantic-to-BEV Transformation**

  Es parteix de la hipòtesi que **una imatge semànticament segmentada obtinguda des de múltiples càmeres instal·lades en un vehicle conté prou informació espacial i contextual per generar una representació precisa de l'escena en format Bird's-Eye View (BEV)** mitjançant models d'aprenentatge profund.

  Aquesta hipòtesi assumeix que el procés de segmentació semàntica prèvia –on cada píxel es classifica segons la seva categoria (carretera, vorera, vehicle, etc.)– proporciona una base suficientment rica i estructurada per inferir la geometria de l'escena des d'una perspectiva superior, fins i tot sense necessitat de reconstrucció 3D explícita ni calibratge geomètric detallat.

- **Simulation-to-Real Generalization**

  La segona hipòtesi estableix que **un model entrenat amb dades provinents d'un simulador d'alta fidelitat, com ara CARLA, pot generalitzar correctament a escenaris simulats diferents i, potencialment, a escenes del món real**.

  Aquest supòsit és clau per reduir la dependència de grans volums de dades reals, sovint difícils d'obtenir i etiquetar. S'assumeix que la variabilitat proporcionada per diferents mapes, escenaris i condicions dins del simulador és suficient per permetre l'aprenentatge de patrons generalitzables, i que les representacions semàntiques redueixen la bretxa entre simulació i realitat.

- **Robustness to Camera Failures**

  Finalment, es proposa la hipòtesi que **el sistema pot continuar generant sortides BEV coherents i útils fins i tot quan una o més entrades de càmera estan absents o degradades**.

  Es considera que, degut a la redundància espacial entre les diferents càmeres i al solapament parcial del seu camp de visió, el sistema pot compensar la manca d'algunes vistes a partir de la informació disponible de la resta de càmeres. Aquesta capacitat de robustesa davant fallades parcials és fonamental per a l'aplicació real del sistema en entorns dinàmics i amb limitacions de maquinari o connectivitat.

Dades

Per entrenar i avaluar el sistema de generació de vistes Bird's-Eye View (BEV), s'ha fet servir el simulador **CARLA**, una eina de codi obert especialment dissenyada per generar entorns urbans realistes orientats a la recerca en conducció autònoma.

Aquest simulador ha permès obtenir de manera controlada grans volums d'imatges etiquetades, reproduint situacions de trànsit diverses i capturant informació visual des de múltiples punts de vista al voltant d'un vehicle en moviment.

Tipus de dades

Les dades capturades es poden classificar en dues modalitats principals:

- **Imatges RGB**: representen escenes del simulador tal com es veurien a través de càmeres convencionals, aportant informació visual detallada com textures, colors i ombres. Cada imatge consta de **3 canals** (RGB).

- **Segmentació semàntica**: es tracta d'imatges en què cada píxel està etiquetat amb una classe semàntica (carretera, vorera, vehicle, edifici, vegetació, etc.). El simulador CARLA ha generat segmentacions amb un total de **30 classes diferents**, que s'han codificat en imatges amb **30 canals** (one-hot encoding), on cada canal representa una classe.

Totes les imatges tenen una resolució inicial de **300 × 300 píxels**

Dimensions i volum

El conjunt de dades generat té un volum aproximat de:

- **30 GB d'imatges RGB** (3 canals)

- **30 GB d'imatges de segmentació semàntica** (30 canals = 30 classes)

Aquestes dades corresponen a un total de **1.046.205 imatges** generades, distribuïdes entre les diferents càmeres i tipus d'informació. Per a cada càmera s'ha guardat tant la seva imatge RGB com la versió segmentada, mitjançant un procés de pretractament previ a l'entrenament.

Per tal d'estalviar còmput innecessari per les imatges de segmentació semàntica, es fa un preprocessament que transforma les imatges a una matriu amb els identificadors de la classe a nivell de píxel.

Configuració de les càmeres

Per tal d'obtenir una visió completa de l'entorn del vehicle, s'ha fet servir una disposició de **8 càmeres al voltant del cotxe**, col·locades a intervals de **45 graus**. Cada càmera cobreix un angle de **120 graus**, de manera que es genera una cobertura visual completa amb **solapament (overlap)** entre càmeres adjacents.

Aquesta configuració permet capturar informació redundant i complementària, essencial per generar una representació coherent de l'escena des d'una perspectiva zenital.

Escenaris i partició

Les dades s'han recollit a partir de **12 recorreguts** diferents amb el vehicle en mode *autopilot* dins el simulador.

Captura dels 12 recorreguts creats a partir del simulador CARLA

Per tal de garantir una avaluació objectiva del rendiment del sistema, s'ha dividit el conjunt de dades en tres subconjunts separats:

- **8 recorreguts** per a entrenament
- **2 recorreguts** per a validació
- **2 recorreguts** per a test

Aquesta divisió assegura que els escenaris utilitzats durant la validació i el test no apareguin durant l'entrenament, fet que permet avaluar la capacitat de generalització del model a entorns desconeguts i comprovar el seu comportament davant situacions noves.

Les dades s'han estructurat en carpetes diferenciades per càmera i per tipus d'informació. A més, també s'han generat els targets de BEV (tant RGB com segmentació semàntica), que

actuen com a ***ground truth*** per entrenar el sistema. La següent és una representació esquemàtica de l'estructura del dataset:

- **Recorregut_N**
    - bev --> Bird's Eye View images (*Ground Truth*)
        - bev_1746196584_568510532.png
        - bev_1746196584_768572568.png
        - ...
    - img1 --> Camera 1 images
        - img1_1746196584_568510532.png
        - img1_1746196584_768572568.png
        - ...
    - imgN --> Camera 2 images
        - imgN_1746196584_568510532.png
        - imgN_1746196584_768572568.png
        - ...
    - ss_bev --> Bird's Eye View semantic segmentation images (*Ground Truth*)
        - ss_bev_1746196584_568510532.png
        - ss_bev_1746196584_768572568.png
        - ...
    - ss_img1 --> Camera 1 semantic segmentation images
        - ss_img1_1746196584_568510532.png
        - ss_img1_1746196584_768572568.png
        - ...
    - ss_imgN --> Camera 2 semantic segmentation images
        - ss_imgN_1746196584_568510532.png
        - ss_imgN_1746196584_768572568.png
        - ...

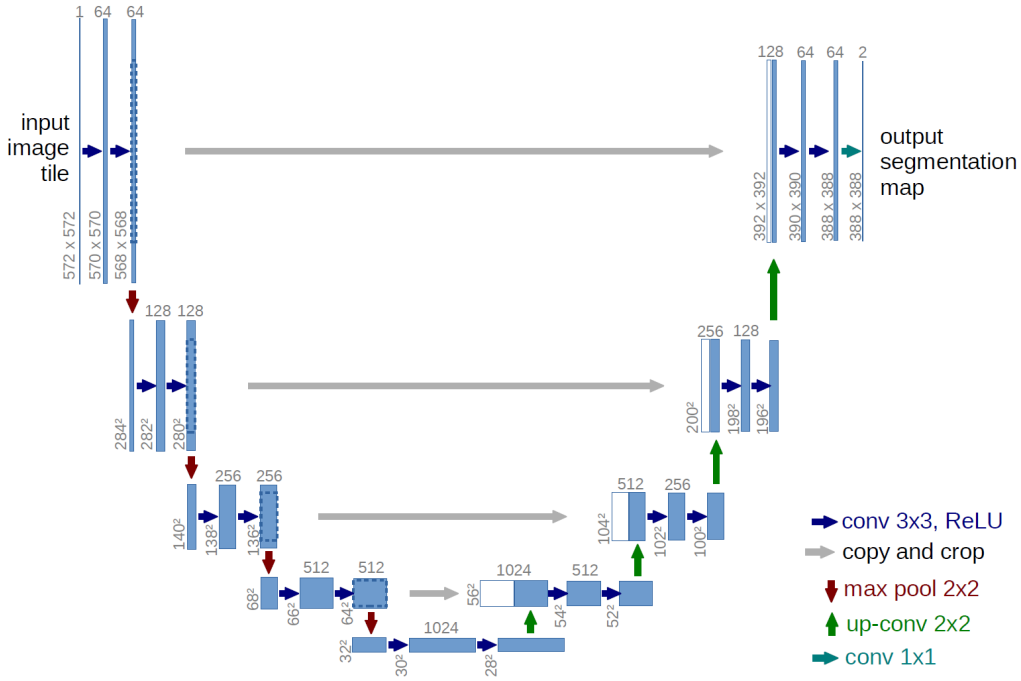
La generació i manipulació de les dades s'ha realitzat tenint en compte la coherència temporal (totes les imatges d'una mateixa mostra comparteixen un timestamp) i l'eficiència d'accés en temps d'entrenament, intentant reduir al màxim la càrrega des del disc.

Arquitectura proposada

Implementació de l'arquitectura

La nostra arquitectura proposada per generar la vista zenital (Bird's-Eye View, BEV) a partir d'imatges captades per múltiples càmeres està basada en una xarxa UNet adaptada. El sistema rep com a entrada 8 imatges RGB o de segmentació semàntica (SS), corresponents a diferents angles de visió al voltant del vehicle.

Cada una d'aquestes imatges és processada per **un encoder individual** basat en convolucions 3×3 i operacions de max-pooling que extreuen característiques rellevants de cada perspectiva. Cal destacar que els encoders no comparteixen pesos en cap moment.



Representació de l'arquitectura d'una UNET

A partir d'aquests 8 encoders, s'obtenen 8 espais latents que representen la informació visual processada de cada càmera.

La clau del nostre model és la **fusió** d'aquests espais latents per crear una representació integrada que sintetitza la informació de totes les vistes. Per combinar la informació provinent de les diferents vistes (una per càmera), s'aplica una **fusió dels espais latents**. Cada encoder processa una imatge i genera una representació (latents). Aquestes representacions es poden combinar de diverses maneres:

- **Mean (mitjana):** Calcula la mitjana de tots els mapes de característiques, donant un resultat suau i equilibrat, útil quan hi ha soroll o variació entre vistes.

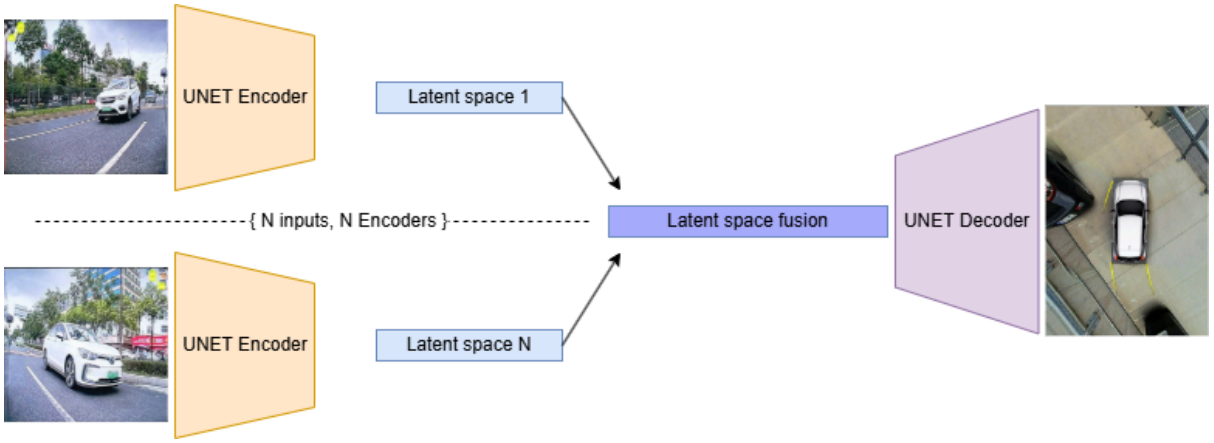
- **Min / Max:** Reté els valors mínims o màxims per posició. Max pot ressaltar les característiques més destacades de cada càmera, mentre que Min pot ser útil en

escenaris específics com la segmentació d'obstacles.

- **Concat (concatenació):** Combina totes les representacions apilant-les en el canal de característiques (dim=1), mantenint tota la informació però incrementant la mida del tensor. Pot millorar el rendiment, però també augmenta la complexitat computacional del decoder.

La clau del nostre model és la **fusió** d'aquests espais latents per crear una representació integrada que sintetitza la informació de totes les vistes. Aquesta fusió es realitza mitjançant diverses estratègies: **la mitjana** (mean), **el mínim** (min), el **màxim** (max) i la **concatenació** (concat) dels espais latents.

Aquestes operacions permeten conservar informació espacial i contextual essencial, assegurant que el model mantingui una representació rica i multidimensional de l'escena.



Esquema de l'arquitectura UNet amb fusió d'espais latents per a la síntesi de la vista BEV

A diferència d'un U-Net clàssic, el nostre decoder rep com a entrada la fusió d'espais latents de múltiples encoders (un per càmera), en lloc d'un sol encoder. Les connexions de salt (***skip connections***) també es fusionen nivell a nivell entre totes les càmeres mitjançant estratègies com la mitjana, el màxim, mínim o la concatenació, segons el mètode escollit. Aquesta fusió per nivell garanteix que la informació espacial capturada per cada vista s'integri de forma coherent en la reconstrucció final. Un cop fusionats, aquests nivells es connecten al decoder amb el mateix esquema jeràrquic i simètric del U-Net, aplicant upsampling i concatenacions amb les característiques corresponents, refinades amb blocs convolucionals.

<u>Avantatges de l'arquitectura</u>

- **Integració de múltiples vistes:** La fusió d'informació provinent de diverses càmeres permet obtenir una visió més completa i precisa de l'entorn, superant les limitacions de les perspectives individuals.

- **Millora de la segmentació:** La combinació de dades de diferents angles facilita la detecció i segmentació d'àrees ocultes o amb poca visibilitat des d'una sola càmera.

- **Escalabilitat:** L'arquitectura és flexible i permet afegir fàcilment més càmeres o perspectives, ampliant la cobertura sensorial del vehicle.

- **Compatibilitat amb diferents tipus de dades:** El model pot treballar tant amb imatges RGB com amb dades de segmentació semàntica prèvia, adaptant-se a diverses fonts d'informació.

Consideracions i inconvenients

És important que la representació latent mantingui més d'una dimensió per conservar la informació espacial i contextual. Això és essencial perquè el model pugui interpretar correctament l'escena i generar una imatge BEV de qualitat.

La selecció del mètode de fusió dels espais latents impacta directament en el rendiment del model. Per exemple, la mitjana i la concatenació aporten diferents formes d'integrar la informació, on la concatenació pot augmentar la dimensionalitat i la riquesa de la representació, mentre que la mitjana o el mínim poden actuar com a operadors d'agregació que redueixen el soroll.

En resum, l'arquitectura UNet adaptada amb la fusió d'espais latents és una solució efectiva per generar una vista zenital integrada a partir de múltiples càmeres, permetent una millor comprensió i representació de l'entorn per a aplicacions de conducció autònoma.

Entrenament

Per a l'entrenament del nostre model de generació de vistes zenitals (BEV), vam comptar amb el suport del **Centre de Visió per Computador (CVC)**, que ens va cedir l'accés a una màquina equipada amb **4 GPUs NVIDIA L40S de 48 GB**. Aquesta infraestructura ens ha permès entrenar models amb una alta demanda de memòria i càlcul de manera eficient. El temps total destinat a la fase d'entrenament ha estat de **37 dies**, considerant els diversos experiments realitzats.

L'entrenament s'ha realitzat **per separat per a cada tipus d'imatge d'entrada**: imatges RGB i imatges de segmentació semàntica. En cap moment s'han compartit pesos entre entrenaments ni s'han combinat funcions de pèrdua. Aquesta separació ens ha permès avaluar el comportament del model de forma específica segons el tipus de representació visual emprada.

Els temps i configuracions per a cada cas han estat els següents:

- **Imatges RGB**:
    - Temps total: **2 hores i 30 minuts**
    - Nombre d'epochs: **50**
    - Amb només 3 canals, el model ha requerit menys memòria, cosa que ens ha permès utilitzar **batch sizes més grans** i accelerar l'entrenament.
- **Segmentació semàntica**:
    - Temps total: **4 hores i 30 minuts**
    - Nombre d'epochs: **25**
    - Aquestes imatges tenen **10 vegades més canals** que les imatges RGB, ja que cada classe de segmentació es representa com un canal independent. Aquesta major dimensionalitat incrementa tant el consum de memòria com la càrrega computacional.
    - Per adaptar-nos a aquesta limitació, vam **reduir el batch size**, fet que implica un entrenament més lent (més iteracions per epoch).

Cerca de paràmetres

Des d'un primer moment vam construir el codi del nostre projecte enfocat en l'execució d'experiments parametritzats, on tenim una funció principal *train()* que rep un conjunt de paràmetres i directoris, on s'escullen totes les característiques de l'execució. A mesura que el projecte avançava, l'arxiu de configuració va anar creixent, amb més flexibilitat i característiques del model, fins arribar a tenir aquest aspecte:

```yaml
experiment: "test-albert-ss"  # Nom de l'experiment

dataset:
 training: [
  'rosbag2_2025_05_08-12_07_12']

 validation: [
  'rosbag2_2025_05_11-17_21_55']

 batch_size: 16
 num_workers: 1
 semantic_segmentation: False
 train_percentage: 1.0 # percentatge de dades de cada mapa que es fa servir
 val_percentage: 1.0

 cameras:
  image_size: [300, 300]  # Per si es vol fer downsize de la imatge en un futur
  num_cameras: 8 # Número de càmeres
  dropout_prob: 0 # Probabilitat de dropout per cada càmera

 bev:
  image_size: [300, 300]  # No es fa servir de moment

architecture:
 model: "unet"
 fusion_method: "mean"  # Opcions: "mean", "max", "min", "concat"

training:
 epochs: 100
 save_interval: 5
 #loss: "CrossEntropy"           # MSE/CrossEntropy
 learning_rate:
  type: "plateau"       # Tipus de scheduler: "step", "plateau" o "cosine"
  initial: 0.00025        # Valor inicial del learning rate (obligatori)
  gamma: 0.5           # Factor de reducció del LR (comú a step/plateau)
  step_size: 10        # (només "step") Cada quantes epochs es redueix el LR
  patience: 5          # (només "plateau") N epochs sense millora abans de reduir LR
  T_max: 50
 loss_components:
  #Dice: 0.5
  #SSIM: 0.2
  #LPIPS: 0.1
  #PSNR: 0.3
  MSE: 1
```

D'aquesta forma, l'arxiu executable *main.py* llegia l'arxiu de configuració i executava l'entrenament a partir d'aquest.

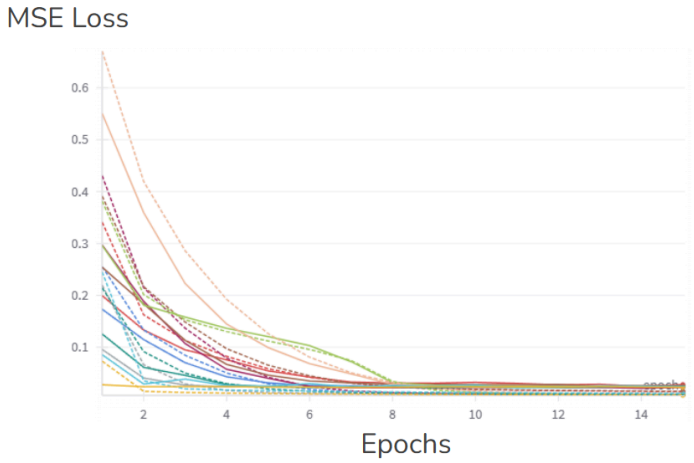
També es va provar la incorporació d'una *Canny Loss,* que penalitzava la diferència entre els contorns detectats (mitjançant l'algorisme de *Canny*) de la predicció i de la imatge *ground truth*. L'objectiu era **reforçar la precisió estructural de les imatges generades, per aconseguir vores més nítides i definides**. Tècnicament, es va implementar aplicant el detector de Canny a ambdues imatges i calculant l'error L1 entre els mapes de contorns resultants.

Amb tantes variables, vam considerar que un bon avanç del projecte seria una execució *gridsearch* per tal d'escollir els millors paràmetres per cada tipus d'imatge (RGB/Segmentació semàntica). Vam escollir les variables que, al nostre criteri, ens donarien més informació de cara a escollir un model base per a seguir experimentant.

Es van estimar els temps d'entrenament totals per al *gridsearch*, per a tenir una mostra suficient d'experiments, i que alhora no ens tregui temps de desenvolupament. Es van monitoritzar les execucions per veure que no es produia overfitting, i el *learning rate scheduler* es comportava de la manera prevista, per evitar aquest.

Després d'un cap de setmana d'execució, vam obtindre els resultats de **72 experiments RGB, i de 52 de Segmentació Semàntica**. En aquest informe s'esmenten algunes de les variables més importants que es van provar en l'execució.

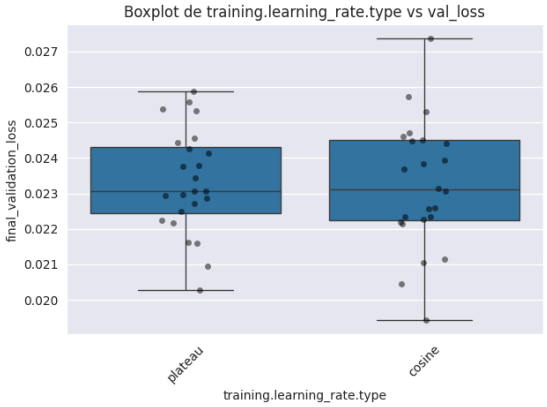
<u>Paràmetres amb imatges RGB</u>

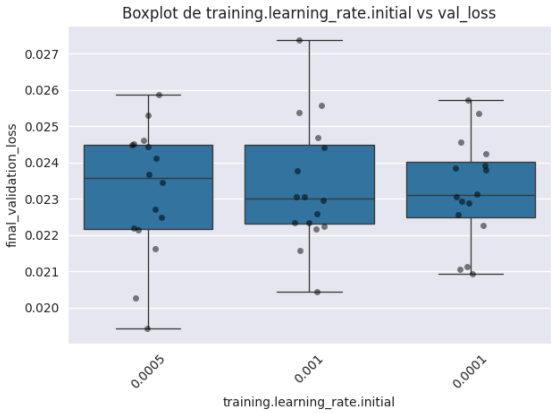


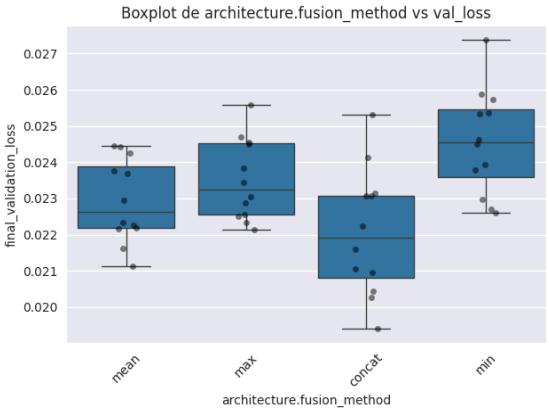
Execucions d'experiments RGB

Pel que fa als experiments amb imatges RGB, s'observa que **no hi ha hagut overfitting**, ja que la *loss* de validació (representada amb línies discontínues) no augmenta en cap moment. Aquest comportament s'aconsegueix gràcies a l'ús del *learning rate scheduler*.

A continuació veiem la comparativa de paràmetres en experiments RGB:







Les conclusions que vam treure a partir d'aquestes gràfiques van ser:

- **El *learning rate* inicial no sembla afectar al rendiment del model**, veiem que no hi ha una relació entre aquest i el valor de loss de validació al final de l'execució.

- El mètode de fusió de l'espai latent que millor funciona és el de **concatenació**, i el que pitjor ho fa és el de **min pooling** (en referència als valors quantitatius).

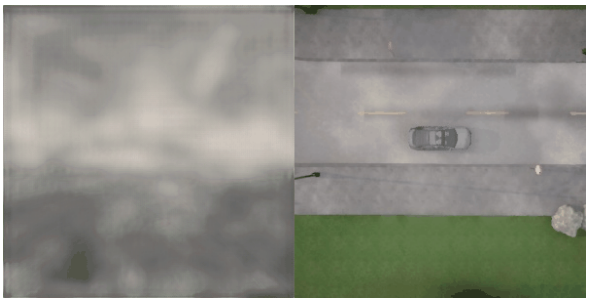
- **El tipus de *learning rate schedules* no sembla afectar a la mitja de la loss de validació final**, però decidim que el millor és el de tipus *plateau*, ja que és adaptatiu a la loss, a diferència del *cosinus*. No es va provar una *learning rate* estàtica ja que, en execucions anteriors veiem que a partir d'un nombre d'epochs començava a fer overfitting, i els *schedulers* eren una solució obvia.

- Al gràfic de losses *MSE* vs *MSE Canny*, veiem com **és major la loss que fa servir *canny map* que l'altra**, que només computa la mitja de la diferència entre el target i la predicció, sense aplicar cap filtre de pesos. És normal que els valors de la MSE Canny siguin majors que a *MSE*, ja que afegeix una penalització a nivell de píxel, incrementant el valor de la loss total.

No hem de comparar els valors de la gràfica per a extreure'n conclusions, sinó observar exemples i jutjar les imatges de prediu el model. No hem volgut normalitzar el resultat de *MSE Canny loss*, ja que donaria menys importància a altres regions de la imatge que poden ser més rellevants per a l'objectiu del projecte, així que vam decidir jutjar nosaltres mateixos els resultats, i vam concloure que no es podia fer una decisió amb resultats qualitatius tan pobres.

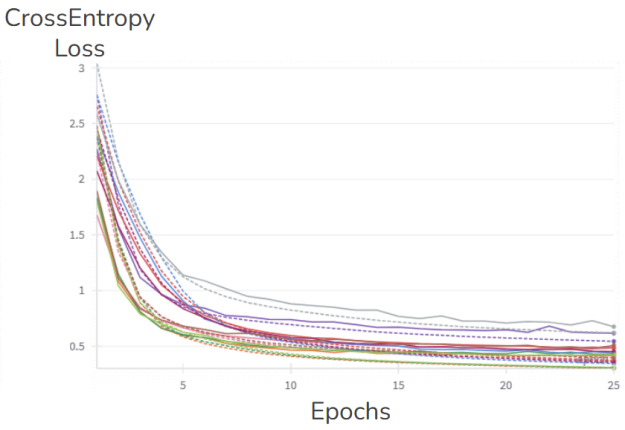
A continuació es mostren els resultats utilitzant com a loss *MSE*, i *MSE Canny*, respectivament:

Resultat utilitzant MSE

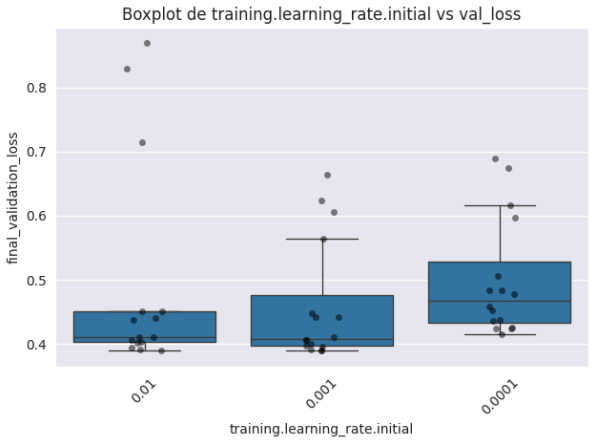


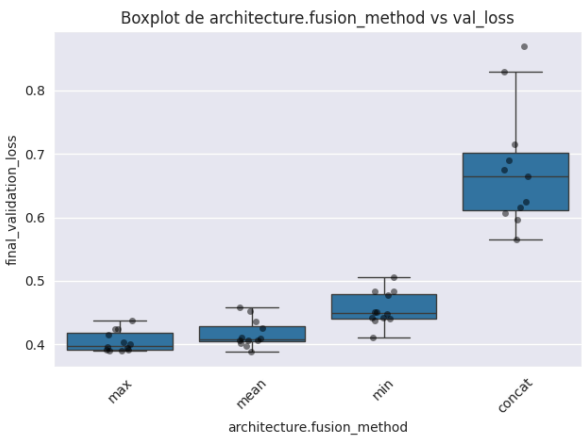
Resultat utilitzant MSE Canny

<u>Paràmetres amb imatges de Segmentació Semàntica</u>



En el cas de segmentació semàntica, veiem com, en general, les corves de loss, tant les de train com les de validation, cauen de forma **més lenta que en el cas de RGB**, i som conscients que hauriem d'haver executat durant més epochs per a extreure conclusions més robustes. Creiem que amb 15 epochs més per experiment podrien haver convergit a valors més baixos encara, però tampoc voliem perdre més temps abans de tenir uns bons paràmetres per començar a fer altres experiments.





En el cas dels experiments de segmentació semàntica, observem tendències que, tot i ser similars a les trobades amb RGB, tenen algunes particularitats pròpies degudes a la naturalesa d'aquestes dades.

- Pel que fa al *learning rate* inicial, veiem a la primera gràfica que **no hi ha una gran diferència entre els tres valors que hem provat**. Tant amb 0.01 com amb 0.001 i 0.0001 obtenim resultats semblants de loss de validació. Destacar, però, que amb valors més baixos, com el 0,0001, la dispersió és una mica menor, fet que podria indicar que el model convergeix de manera més estable, a diferència que amb el valor 0,01, on es veuen clarament outliers. No obstant això, **no observem una relació clara entre el valor inicial del *learning rate* i la loss final de validació**.

- En el cas del mètode de fusió dels espais latents, veiem diferències més significatives. A la gràfica es pot veure com *mean* i *max* funcionen força bé, amb valors de loss baixos i molt poca dispersió. Seguidament, *min*, té resultats correctes, però pitjors que *mean* i *max*, per tant, no destaca especialment. Per últim, amb el mètode *concat*, els resultats són clarament pitjors, tant per la mitjana com per la dispersió i la variabilitat. Això pot ser degut al fet que aquest mètode afegeix molta complexitat al model, ja que apila tota la informació de les càmeres sense reduir-la, i pot fer que li costi molt més generalitzar a causa del soroll.

  Per tant, si haguéssim de triar, definitivament **ens quedaríem amb *mean* o *max* com a mètodes més estables per treballar amb segmentació semàntica**.

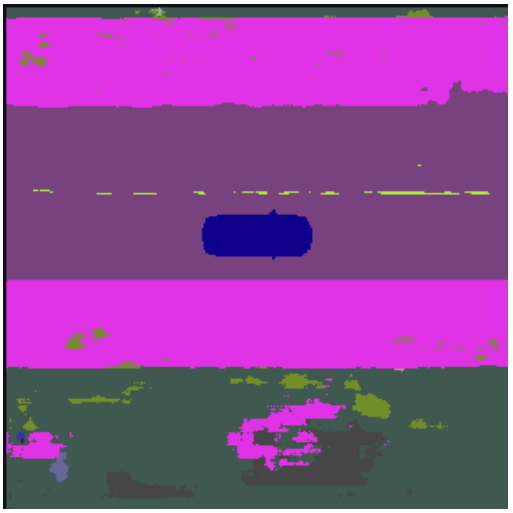
- Pel que fa al tipus de *scheduler* per al *learning rate*, el gràfic **no mostra una diferència significativa entre *plateau* i *cosine***. Tots dos ofereixen una evolució estable i similar respecte a la loss de validació. Tot i així, continuem amb la preferència pel scheduler *plateau*, ja que permet una adaptació directa a la loss, reduint el *learning rate* només quan la millora es para, mentre que *cosine* segueix una pauta fixa.

Experiments

Un cop vam tenir totes les dades preparades, i després de l'exploració de paràmetres per identificar les millors configuracions, tant a nivell de loss de validació com de qualitat visual de les sortides, vam realitzar diversos experiments per a provar les nostres hipòtesis.

En el primer experiment vam voler fer una **comparativa exhaustiva entre imatges RGB i de Segmentació Semàntica**. Per a fer aquesta comparativa, es va triar, és clar, els experiments que donaven millor resultats, tant qualitativament com quantitivament.

RGB SSIM



Segmentació Semàntica

En el cas de RGB, s'aprecia una sortida més natural pel que fa a textures i intensitat, però força borrosa i amb poca nitidesa. Els límits entre calçada, vorera i vehicle no estan clarament definits, i el soroll fa que la interpretació global sigui més difícil.

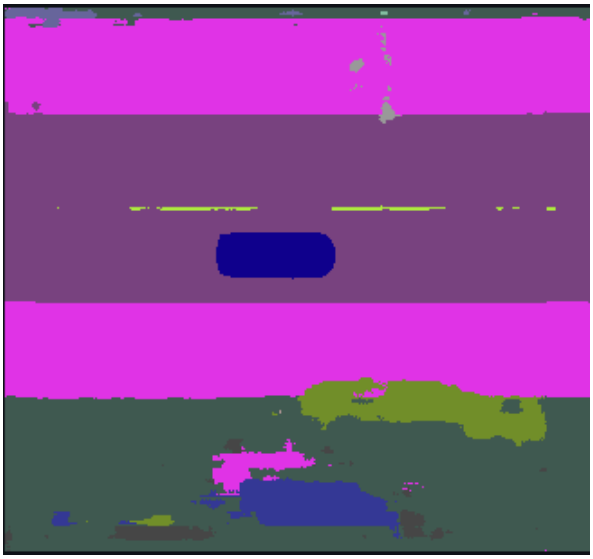
En canvi, amb segmentació semàntica, la predicció mostra una estructura molt més clara. Es poden identificar fàcilment les línies de la carretera, el vehicle i les àrees d'entorn. El fet de codificar-ho per classes (30 canals one-hot) ajuda el model a entendre millor l'estructura de l'escena i a reconstruir una vista zenital més coherent, on entenem molt millor l'entorn.

Per tant, aquests resultats ens fan pensar que, per a un objectiu com és en el nostre cas, la conducció autònoma, **les prediccions obtingudes de segmentació semàntica són molt més útils i estructurades que les de RGB**. Ara bé, cal tenir en compte que aquest avantatge es dona en un entorn simulat on les imatges semàntiques estan perfectament etiquetades. A la vida real, aquest tipus d'entrada seria més difícil, per no dir impossible, d'obtenir amb la mateixa qualitat, fet que evidentment **aquests resultats no són extrapolables a un entorn real**.

En segon lloc, un dels experiments que vam voler provar va consistir en **avaluar la robustesa del sistema davant possibles fallades**, com ara la pèrdua d'alguna càmera durant l'execució. Aquest escenari és molt realista en contextos de conducció autònoma, on pot haver-hi càmeres espatllades, tapades o directament desconnectades.

Per simular aquest comportament, vam fer dues aproximacions diferents. D'una banda, vam entrenar models amb *dropout* aplicat a les càmeres d'entrada, per tal que aprenguessin a reconstruir la vista BEV tot i tenir vistes incompletes de forma aleatòria. De l'altra, vam entrenar un model completament amb només 4 càmeres en comptes de 8, orientades al nord, sud, est i oest relatiu al vehicle, és a dir, descartant les càmeres orientades a les cantonades del cotxe (diagonals).

Els resultats van ser sorprenents. En les prediccions fetes amb només 4 càmeres, la qualitat final no només **no empitjora de manera clara**, sinó que en alguns casos és **fins i tot millor que amb les 8 càmeres originals**. A les següents imatges es veuen dues sortides, corresponents a les prediccions fetes a partir de 8 i 4 càmeres, respectivament.



Predicció amb 8 càmeres

Predicció amb 4 càmeres

Aquests resultats ens fa pensar que el model, quan treballa amb totes les càmeres, pot estar rebent informació redundant o fins i tot contradictòria (per angles solapats), cosa que dificulta l'aprenentatge.

Amb només 4 càmeres, el problema es simplifica i el model pot aprendre una relació més directa entre entrada i sortida. Això no vol dir que menys càmeres siguin sempre millors, però en el nostre cas, sí que obre la porta a explorar arquitectures més lleugeres o configuracions òptimes de càmeres que maximitzin la informació amb el mínim cost computacional.

Com a tercer experiment ens vam centrar en una proposta que va sorgir durant una de les reunions de seguiment del projecte que realitzàvem setmanalment amb el CVC. Es va comentar que, més enllà de l'arquitectura clàssica basada en U-Net, es podria plantejar **l'integració de transformers per millorar la predicció**, concretament quan es treballa amb imatges RGB.

La motivació d'aquest es basava en el fet que els transformers, a diferència de les convolucions tradicionals, poden captar relacions espacials mitjançant mecanismes d'attention, i consideràvem que això podria ajudar a reconstruir millor la vista BEV.

Per implementar-ho, vam mantenir la part d'encoder basada en U-Net (que ja teníem desenvolupada), però la part del decoder, la vam substituir per una basada en transformer encoders. Aquest nou mètode rep l'espai latent fusionat i aplica auto-attention per sintetitzar la sortida. El nostre objectiu era veure si aquest canvi ajudaria a millorar la qualitat general o la definició de les imatges generades.

A continuació, veiem el resultat d'aquest model de U-Net i transformers:

Predicció utilitzant Unet+Transformers

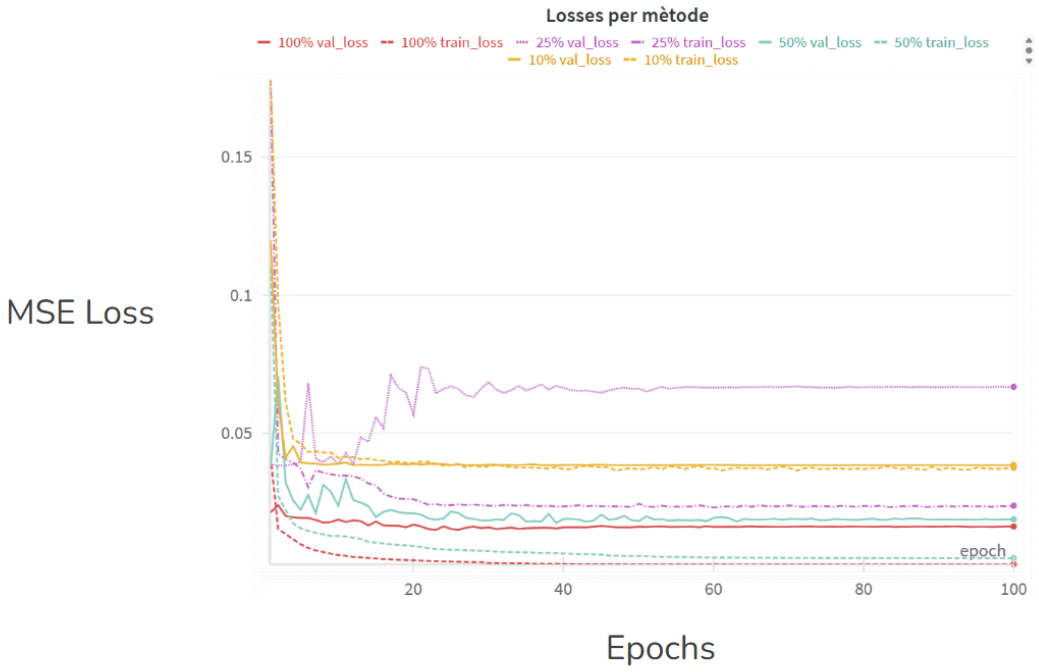
Tot i que no hem pogut fer una comparació exhaustiva amb molts resultats quantitatius, les primeres proves visuals són prometedores. Es percep **una millora lleu en la nitidesa general i sobretot en la localització del vehicle**, i sembla que **el model entén millor la jerarquia espacial de l'escena**, és a dir, manté els objectes al seu lloc.

Caldria més entrenament, ajustar hiperparàmetres, i segurament refinar el model però és una línia molt interessant a continuar explorant.

Com a darrer experiment, es va analitzar **com afecta al rendiment del model la quantitat de dades utilitzada durant l'entrenament**. La nostra hipòtesi inicial era que un augment en la quantitat de dades permetria al model generalitzar millor, i per tant, obtenir millors resultats.

Per provar-ho, vam entrenar diversos models amb el mateix escenari de paràmetres, però fent servir **quatre percentatges** diferents del dataset total: 10%, 25 %, 50 % i 100 % de les dades totals. Tots els models es van entrenar amb imatges RGB i amb 100 epochs, per tal de mantenir les condicions comparables.

Les següents imatges mostren l'evolució quantitativa de la loss en funció de la mida del dataset:



Execucions d'experiments RGB amb diferents percentatges del dataset total

Tal com s'observa al gràfic, a mesura que augmenta el percentatge del conjunt de dades utilitzat durant l'entrenament, el model mostra un millor rendiment, reflectit en una *loss* més baixa. El millor resultat s'obté utilitzant el 100% del dataset, la qual cosa indica que, molt probablement, disposar de més dades permetria al model generalitzar millor i obtenir un rendiment superior.

Predicció (100% dataset)

Predicció (50% dataset)

A nivell visual, la diferència és molt més notable. Amb el total de dades (imatge de l'esquerra), el model és capaç de reconstruir l'escena, tot i que clarament es veu borrosa i amb soroll, podem distingir la calçada, el vehicle i elements de l'entorn. Amb el 50 % (imatge de la dreta), la predicció és més borrosa, però raonablement coherent i s'arriba a apreciar l'entorn.

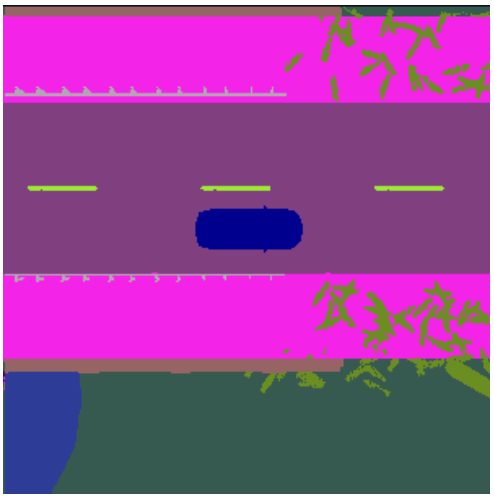
Aquests resultats ens validen que, en un problema com aquest, tenim prou dades per realitzar una predicció que entengui parcialment l'entorn, però que **no són suficients per obtenir una reconstrucció precisa i robusta**, sobretot quan l'objectiu és un model segur per a la conducció autònoma. Tot apunta que, tractant-se d'una tasca de generació d'imatges, **necessitaríem un volum molt més gran de dades** per tal que el model fos capaç de capturar millor la varietat d'escenes (towns) possibles i produir sortides de qualitat més alta.

Resultats

En aquest projecte no ens hem centrat a obtenir un únic resultat final, sinó que l'objectiu ha estat explorar una àmplia varietat de configuracions, paràmetres i arquitectures per a entendre millor com diferents factors afecten el rendiment del model en la generació de vistes BEV. S'ha fet una anàlisi exhaustiva de diverses combinacions de tipus d'imatge (RGB vs. segmentació semàntica), nombre de càmeres d'entrada, mètodes de fusió d'espais latents, schedulers de learning rate i funcions de pèrdua. Aquest enfocament experimental ens ha permès identificar quines opcions ofereixen un millor compromís entre qualitat, eficiència i complexitat.

D'entre tots els models provats, el que ha demostrat millor comportament, i el que volem destacar com a "model estrella" ha estat el que utilitza imatges de **segmentació semàntica** com a entrada, amb **4 càmeres**, i fusió per mitjana (**mean**) dels espais latents. Aquest model manté una qualitat visual molt propera a la del model entrenat amb 8 càmeres, però redueix significativament tant el temps d'entrenament (aproximadament a la meitat), cosa que el fa més viable per a entorns reals o sistemes amb limitacions de recursos.

Predicció



Target

Els resultats qualitatius mostren que la reconstrucció és coherent, clara i útil per a tasques de comprensió de l'escena senzilla. Això suggereix que la redundància de càmeres pot aportar més soroll que informació útil.

Per tant, tot i haver explorat molts models i configuracions, presentem aquest model com el millor resultat obtingut, fruit del procés iteratiu d'experimentació i anàlisi detallada. Aquest model representa el millor equilibri entre qualitat, temps d'entrenament i simplicitat, i marca un bon punt de partida per a treballs futurs més enfocats a l'optimització i la generalització del sistema.

Inconvenients

Tot i els avantatges de la nostra arquitectura basada en U-Net per a la generació de vistes zenitals a partir de múltiples càmeres, el desenvolupament i entrenament del sistema presenta diversos **inconvenients tècnics i pràctics** que cal tenir en compte:

- **Risc d'overfitting:** Donada la mida limitada del conjunt de dades, el model tendeix a ajustar-se massa a les mostres d'entrenament, perdent capacitat de generalització. Per mitigar aquest efecte, s'ha implementat una estratègia de **learning rate schedule** per controlar millor l'actualització dels pesos durant l'entrenament.

- **Dependència de les càmeres:** El sistema depèn fortament de la disponibilitat de totes les vistes d'entrada. En escenaris reals, és habitual que algunes càmeres fallin o quedin parcialment ocultes. Per això, hem experimentat amb ***dropout* de càmeres**, simulant la pèrdua d'entrada en alguna vista per forçar el model a ser més robust.

- **Alt consum computacional:** Entrenar models amb múltiples càmeres i imatges d'alta resolució implica una elevada càrrega de treball per a la GPU. Per fer-ho més viable, s'ha optat per **preprocessar les dades** i optimitzar l'arquitectura per reduir al màxim el temps i el cost computacional durant l'entrenament.

- **Escassetat de dades per a una tasca de generació:** La generació d'imatges (com és la construcció d'una vista zenital coherent) és una tasca complexa que normalment requereix grans quantitats de dades per entrenar-se de manera efectiva. En el nostre cas, disposem de molt poques mostres, fet que **limita la capacitat d'aprenentatge del model** i restringeix la seva aplicabilitat a entorns més realistes.

Conclusions i millores a futur

Tot i que el sistema desenvolupat ha estat capaç de generar vistes zenitals a partir d'imatges captades des de múltiples càmeres, som conscients que **no hem assolit la millor solució possible**. Aquest fet no és fruit de cap error, sinó d'un conjunt de limitacions tècniques i metodològiques que afecten directament la qualitat del resultat.

Limitacions i causes

La nostra arquitectura basada en U-Net ha mostrat un comportament raonablement efectiu dins l'entorn simulat, però **no ha assolit una representació detallada i robusta de l'entorn**. Els principals motius d'aquest resultat són:

- **Manca de dades suficients:** Perquè una U-Net sigui realment eficaç en una tasca tan complexa com la generació d'una vista BEV fusionant múltiples perspectives, es necessita una gran quantitat de dades variades i realistes. Amb el conjunt de dades actual, estimem que **necessitaríem com a mínim 100 vegades més dades** per entrenar un model amb una generalització sòlida.

- **Simplicitat de l'arquitectura:** Tot i que la U-Net és una base potent, la seva estructura estàndard pot ser insuficient per a capturar les relacions espacials complexes que es donen entre les diferents vistes d'una escena tridimensional.

Millores a futur

De cara a futures iteracions del projecte, identifiquem diverses línies de millora clau:

- **Increment del volum de dades**: Augmentar massivament la quantitat de dades, idealment amb entorns més realistes o amb simulacions més variades, per tal de millorar la capacitat d'aprenentatge del model.

- **Explorar arquitectures alternatives**: Provar altres enfocaments que permetin una millor fusió de la informació multivista, com ara:

    - Transformers espacials per integrar millor les vistes (una millor arquitectura).

    - Arquitectures híbrides CNN-Transformer.

- **Aplicació de tècniques adversàries**: Incorporar un **discriminador en el context d'un GAN (Generative Adversarial Network)** per tal de guiar el model generador a produir imatges més realistes i coherents des del punt de vista estructural.

- **Estudi de tècniques de regularització i augment de dades**: Per combatre l'escassetat de dades i millorar la robustesa del model davant escenaris nous o incomplets (com fallades de càmeres).

En resum, aquest projecte ha representat un primer pas sòlid en la direcció correcta, tot i que encara lluny d'una solució òptima. Hem establert una base clara sobre la qual construir futures millores tant a nivell d'arquitectura com de dades i mètodes d'entrenament.

Annexos

Annex A. Repositori del projecte

El codi complet del projecte es troba disponible al següent repositori de GitHub:

https://github.com/XarNeu-EngDades/project24-25-11

Aquest repositori conté tot el desenvolupament del model fins a la data de la seva presentació. Inclou el codi principal, les arquitectures del model, els mòduls creats i les configuracions.

Es destaca també, dins del repositori , la pàgina showcase on es troben GIFs d'inferència del model, i enllaços a informes de Weights & Biases (wandb), per a més detalls dels experiments fets que no s'han inclòs en aquest informe.