UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escola Tècnica Superior d'Enginyeria Informàtica

Aprenentatge profund per a la generació automàtica de màscares de zones d'interès en imatges de carrosseries de cotxe

Treball Fi de Grau

Grau en Enginyeria Informàtica

AUTOR/A: Descals Carbonell, Vicent

Tutor/a: Juan Císcar, Alfonso

Cotutor/a extern: MAHIQUES SIFRES, XAVIER

CURS ACADÈMIC: 2021/2022

Resumen

La inteligencia artificial (IA) es uno de los campos más activos de la informática actual, ofrece un amplio rango de aplicaciones al mundo real, desde las recomendaciones que nos sugieren plataformas como YouTube, hasta grandes aplicaciones Industriales. En este caso, dentro del sector industrial, el uso de la IA suele estar enfocada a la automatización de procedimientos que resultan muy costosos para los trabajadores.

Se propone un Trabajo Fin de Grado (TFG) centrado en uso de la IA para la segmentación semántica de imágenes con el uso de redes neuronales profundas. Estos modelos se quieren aplicar al sector del automóvil y, en particular, a imágenes de carrocerías de coche. Se pretende generar automáticamente máscaras de zonas de interés para la detección posterior de defectos. Se empleará un conjunto grande de imágenes, con máscaras hechas a mano, para entrenar las redes. La implementación se llevará a cabo con Sublime Text, Anaconda y bibliotecas de Pytorch especializadas en tareas de visión artificial y aprendizaje profundo. Los resultados obtenidos por las redes aprendidas se evaluarán con los niveles de precisión que requiere la tarea. Se prevé que el sistema ayude a automatizar y acelerar un procedimiento que se hace ahora de forma manual por un técnico, con un gran coste temporal.

Palabras clave: Inteligencia artificial, aprendizaje profundo, redes neuronales, automóviles, carrocerías de coches, detección de defectos.

Resum

La intel·ligència artificial (IA) és un dels camps més actius de la informàtica actual, ofereix un ampli rang d'aplicacions al món real, des de les recomanacions que ens suggereixen plataformes com YouTube, fins a grans aplicacions industrials. En aquest cas, dins del sector industrial, l'ús de la IA sol estar enfocada a l'automatització de procediments que resulten molt costosos per als treballadors.

Es proposa un Treball de FI de Grau (TFG) centrat en l'ús de la IA per a la segmentació semàntica d'imatges amb l'ús de xarxes neuronals profundes. Aquests models es volen aplicar al sector de l'automòbil i, en particular, a imatges de carrosseries de cotxe. Es pretén generar automàticament màscares de zones d'interès per a la detecció posterior de defectes. S'emprarà un conjunt gran d'imatges, amb màscares fetes a mà, per tal d'entrenar les xarxes. La implementació es durà a terme amb Sublime Text, Anaconda i biblioteques de Pytorch especialitzades en tasques de visió artificial i aprenentatge profund. Els resultats obtinguts per les xarxes apreses s'avaluaran amb els nivells de precisió que requereix la tasca. Es preveu que el sistema ajude a automatitzar i accelerar un procediment que es fa de forma manual per un tècnic, amb un gran cost temporal.

Paraules clau: Intel·ligència artificial, aprenentatge profund, xarxes neuronals, carrosseries de cotxe, automòbils, detecció de defectes.

Abstract

Artificial intelligence (AI) is one of the most active fields in computer science today, offering a wide range of real-world applications from computing today, offering a wide range of real-world applications, from the recommendations suggested by platforms such as YouTube, to large industrial applications. In this case, within the industrial sector, the use of AI is often focused on the automation of procedures that are very costly for workers.

A Final Degree Project (TFG) is proposed that focuses on the use of AI for semantic image segmentation using deep neural networks. These models are to be applied to the automotive sector and, in particular, to images of car bodies. The aim is to automatically generate masks of areas of interest for subsequent defect detection. A large set of images, with hand-made masks, will be used to train the networks. The implementation will be carried out with Sublime Text, Anaconda and Pytorch libraries specialised in computer vision and deep learning tasks. The results obtained by the learned networks will be evaluated with the levels of accuracy required by the task. It is expected to help automate and speed up a procedure that is done manually by a technician, with a high time cost.

Keywords: Artificial intelligence, deep learning, neural networks, car bodies, cars, defects detection.

Tabla de continguts

1. Introducció

La intel·ligència artificial (IA) és un dels camps més actius en la informàtica actual. S'inclou dins de la rama de les ciències de la computació i té aplicacions en diversos sectors com poden ser l'industrial, mèdic o comunicació.

Al camp de la intel·ligència artificial trobem els camps de l'aprenentatge automàtic i l'aprenentatge profund, entre altres. A pesar de que les seues aplicacions poden donar-se en entorns completament distints, sempre tenen una mateixa base comú a partir de la qual funcionen. Aquests sistemes es basen en diferents arquitectures de xarxes neuronals, que poden ser més o menys profundes depenent dels requisits de la tasca a realitzar. Aquestes xarxes seran les encarregades d'aprendre i identificar patrons a les dades amb les quals treballen en cada cas.

Vista la creixent evolució de les tècniques d'intel·ligència artificial i els bons resultats aconseguits, es necessari començar a aplicar aquestes ferramentes a sectors en els qual hi han tasques manuals que poden ser realitzades de forma més ràpida i eficient per sistemes d'aprenentatge profund, en aquest cas concret, al sector automobilístic.

Les tècniques de IA aplicades a l'automobilisme, com la detecció de micro-defectes i generació automàtica de les màscares de detecció corresponents, no se'ls ha donat la suficient importància i per tant no s'han invertit els recursos suficients per al seu correcte desenvolupament, i per tant tenim com a resultat processos manuals i repetitius realitzats per usuaris i no per sistemes de forma automàtica.

Aleshores, a aquest treball de fi de grau es planteja una solució al problema descrit anteriorment, la generació automàtica de màscares per a detecció de micro-defectes en carrosseries de cotxes. La finalitat és poder realitzar aquestes tasques repetitives d'una forma més ràpida, eficient i automàtica.

1.1 Motivació

Donada la situació exposada anteriorment, s'ha vist que el desenvolupament d'un sistema d'aprenentatge profund amb capacitat de generar màscares de detecció de forma automàtica es necessària per a

l'optimització de la cadena de producció de les grans factories com ho són les de l'automòbil, on es busca ser el més eficient possible.

A dia d'avui aquest es un procés que realitza un tècnic de forma completament manual i que necessita una gran quantitat de temps per tal de completar-se. Amb la implantació d'un sistema com el descrit anteriorment aconseguiríem el mateix resultat o millor que l'aconseguit fins ara, però amb una reducció més que considerable del temps emprat en aquesta tasca.

Resumint, la creació d'un sistema d'aprenentatge profund per a la generació automàtica de màscares de detecció està motivada primerament per la inquietud personal per l'ús de tècniques d'intel·ligència punteres a un sector industrial que es està en un desenvolupament continu, com es el sector automobilístic. Per un altre costat, per substituir un procediment manual, costos i repetitiu que amb les tècniques existents d'avui en dia no té sentit que es segueixen fent d'aquesta manera. Tots aquests motius comporten a la realització d'un projecte complet i plenament funcional.

1.2 Objectius

L'objectiu bàsic d'aquest Treball Final de Grau (TFG) és la creació d'un sistema d'aprenentatge profund per a la generació automàtica de màscares de detecció que transforme un procés manual, costós i repetitiu en un automàtic i més eficient.

Per acomplir aquest objectiu principal, el treball d'aquest TFG s'ha organitzat al voltant de 3 objectius principals:

- Ampliar formació sobre el procés de generació de màscares de detecció, passos a seguir, avantatges i limitacions.

- Estudiar aplicacions de segmentació semàntica aplicades a carrosseries de cotxes i treballar amb imatges d'entrada de gran resolució. Explicar el model d'aprenentatge profund basat en xarxes neuronals amb una arquitectura U-Net.

- Fer proves d'aquest model amb tasques bàsiques per comprovar el seu correcte funcionament, i avaluar el seu rendiment amb dades reals.

1.3 Estructura del document

El desenvolupament d'aquest Treball de Fi de Grau (TFG) va a constar de les següents parts:

- El capítol 2 explica el procés manual que es realitza actualment amb les imatges de carrosseries de cotxes. Es detallarà quines són les dades de les que es disposen per a la realització del projecte i com s'organitzen. Per finalitzar aquesta part, una introducció a les tasques de segmentació semàntica d'imatges.

- El capítol 3 explica en què consisteixen las xarxes U-Nets i la seua aplicació per a tasques de segmentació semàntica. Es detallaran les característiques i elements que conformen el sistema, de igual forma que els primers experiments realitzats, amb resultats i mètriques dels mateixos.

- El capítol 4 descriu el disseny i implementació de la solució escollida després d'haver realitzat l'anàlisi corresponent. Explicació detallada de la metodologia utilitzada i dels passos seguits per tal d'acomplir els objectius del TFG.

- El capítol 5 recull conclusions sobre el treball realitzat, propostes per a futures ampliacions i relació del treball amb la matèria cursada durant el grau d'enginyeria informàtica.

2. Preliminars

Abans d'entrar en detalls tècnics i sobre la implementació del sistema anem a descriure el procediment industrial dins del qual es troba el producte d'aquest TFG.

Dins d'una factoria automobilística podem trobar infinitat de processos que conformen la fabricació de cotxes, en aquest cas anem a parlar sobre el procediment encarregat d'identificar xicotets defectes a les carrosseries de cotxes després del procés de pintura. Aquest procediment consta d'un túnel format per arcs de llum LED i diverses càmeres repartides per tot el túnel amb la finalitat de capturar totes les parts del cotxe sense excepció. El que capturen aquestes càmeres es la reflexió que produeix la llum emesa pels arcs sobre la carrosseria del cotxe.

A partir d'aquestes imatges s'apliquen algoritmes de visió artificial com puguen ser per exemple càlculs d'ombralls automàtics, fusions de màxims o fusions de diferències. Per poder aplicar alguns d'aquests algoritmes cal identificar a cada imatges quina zona de la carrosseria cal inspeccionar en busca de defectes, ací és on entrarà en funcionament el nostre sistema d'aprenentatge profund per a la generació automàtica de màscares de zones d'interès en imatges de carrosseries de cotxe.

2.1 Segmentació manual de carrosseries de cotxe

Com la gran majoria de processos industrials als seus inicis es realitzava de forma manual i amb el temps han passat per un període d'automatització en qual s'ha substituït el treball dels operaris per sistemes informàtics o industrials que realitzaven les tasques d'una manera més eficient i precisa.

El mateix passa amb la generació de màscares que ens delimiten les zones a inspeccionar a les imatges de les carrosseries dels cotxes, diferenciant entre el que pertany a la carrosseria, el que es el fons o zona no inspeccionable com per exemple les zones destinades a retrovisors, finestres o plàstics protectors. Aquest procediment s'encarregava un operari que dissenyava una màscara especifica per a cada càmera de cada model de cotxe distint, tenint en compte que s'utilitzen una mitja d'entre 28-30 càmeres per inspeccionar els cotxes que varia depenent de la factoria, ens podem fer una idea del cost temporal d'aquesta tasca que podia portar fins dues setmanes completar-la.

A la Figura 1 es poden veure alguns exemples.

Figura 1. Fusió de diferències (esquerra), màscara (mig), superposició de la màscara a la fusió (dreta)

2.2 Conjunt de dades disponibles

Per a la realització d'aquest projecte es disposa d'un conjunt de dades format per imatges extretes de la factoria de Volkswagen i altres de la factoria de Mercedes-Benz. Dins de les imatges distingim dos tipus d'imatges, les corresponents a les fusions de diferències i les imatges que representen les màscares associades a cadascuna de les imatges anteriors.

El conjunt total, es a dir, les imatges de les dues factories juntes, es de 99018 imatges (fusions + màscares). Si vegem quantes pertanyen a cada factoria la distribució queda de la següent forma:

- Volkswagen: 77578 (fusions + màscares)
- Mercedes-Benz: 21440 (fusions + màscares)

Com podem veure es disposen de moltes més imatges de Volkswagen que de Mercedes, concretament Volkswagen representa un 78.34% del conjunt total de dades i Mercedes un 21.66%.

Als experiments que es detallaran a capítols posteriors s'utilitzaran dos distribucions de dades diferents del conjunt complet. Un d'ells serà únicament amb el conjunt complet d'imatges de Volkswagen, i l'altre amb el conjunt complet de les dues factories.

Passem a detallar el primer dels conjunts de dades:

El conjunt complet es de 77578 imatges com havíem mostrat abans. Aquestes dades anem a dividir-les en 3 subconjunts anomenats *train, val* i *test.* La distribució queda així:

- *Train:* 70% corresponent als cotxes amb data de producció més antiga
    - Fusions: 27018
    - Màscares: 27018
- *Val:* 15% corresponent als cotxe més nous que els de *train* i més vells que els de *test.*
    - Fusions: 5832
    - Màscares: 5832
- *Test:* 15% corresponent als últims cotxes produïts.
    - Fusions: 5939
    - Màscares: 5939

El segon conjunt està format tant per imatges de la factoria de Volkswagen com de la de Mercedes. De igual forma que en el cas anterior farem una divisió en tres subconjunts de *train, val* i *test.* La distribució queda així:

- *Train:* 70% corresponent als cotxes amb data de producció més antiga
    - Volkswagen:
        - Fusions: 26960
        - Màscares: 26960
    - Mercedes:
        - Fusions: 7504

        - Màscares: 7504
- *Val:* 15% corresponent als cotxe més nous que els de *train* i més vells que els de *test.*
    - Volkswagen:
        - Fusions: 5778
        - Màscares: 5778
    - Mercedes:
        - Fusions: 1608
        - Màscares: 1608
- *Test:* 15% corresponent als últims cotxes produïts.
    - Volkswagen:
        - Fusions: 5776
        - Màscares: 5776
    - Mercedes:
        - Fusions: 1608
        - Màscares: 1608

Es pot veure que aquest conjunt suma 98468 imatges i no les 99018 que havíem dit anteriorment, això es degut a que s'han descartat unes poques per tractar-se d'imatges defectuoses.

2.3 Segmentació semàntica d'imatges

La tasca de generar màscares de zones d'interès en imatges de carrosseries de cotxes pot identificar-se dins del camp de visió artificial com una tasca de segmentació que consisteix en dividir els píxels d'una imatge en varies regions denominades segments. La segmentació és un procés de classificació per píxel que assigna una categoria o classe a cada píxel de la imatge analitzada. Aquesta tasca general pot dividir-se en tasques més especialitzades com per exemple.

- Reconeixement d'imatges
- Detecció d'objectes
- Segmentació d'instàncies
- Segmentació semàntica

A la Figura 2 vegem una representació de la funcionalitat d'aquestes tasques.

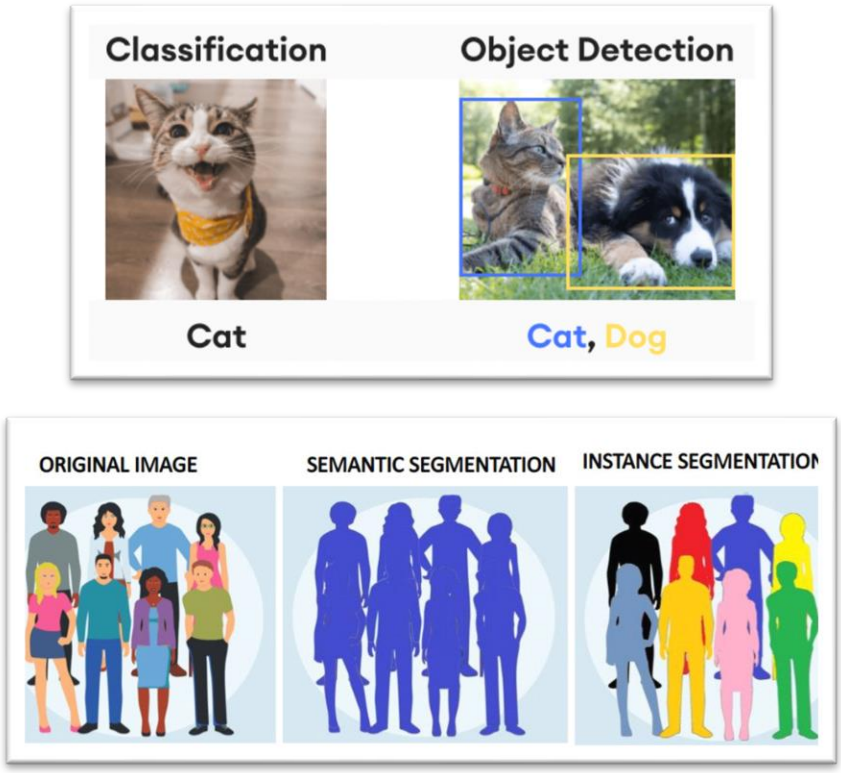


Figura 2. Il·lustració diferències entre segmentació semàntica i d'instàncies, reconeixement d'imatges i d'objectes.

Aquestes quatre tasques estan totes elles enfocades al tractament d'imatges, però cadascuna amb una finalitat diferent. El reconeixement d'imatges busca assignar una etiqueta a la imatge completa, mentre que la detecció d'objectes localitza diferents objectes dins d'una imatge. La segmentació d'instàncies tracta els múltiples objectes de la mateixa classe com a objectes individuals, mentre que la segmentació semàntica agrupa tots aquells que pertanyen a la mateixa classe. La tasca de segmentació d'instàncies resulta molt més complexa que la segmentació semàntica, una de les seues aplicacions més útils és contar el número d'objectes d'una mateixa classe que apareixen a una imatge.

Podem ordenar aquestes tasques segons el nivell de detall que ens aporten, el reconeixement d'imatges és aquella que ens aporta una visió general de la imatge, la detecció d'objectes entra un poc més al detall localitzant el contingut de la imatge, la segmentació semàntica ens permet un nivell més alt de precisió definint formes i fronteres dels objectes de la imatge. I per finalitzar la segmentació d'instàncies ofereix un major detall i precisió que les tres anteriors però també té un cost computacional més elevat.

Anem a centrar-nos amb la segmentació semàntica, que es la que utilitzarem per al desenvolupament del nostre sistema de generació automàtica de màscares. Algunes de les seues aplicacions que podem trobar en diferents camps de la informàtica són:

- **Conducció autònoma:** Per a que un sistema d'aquest tipus funcione, es necessari fer un anàlisis de l'entorn de forma molt precisa, identificant altres vehicles, persones, semàfors o qualsevol element que hi haja a la carretera i als voltants.

- **Inspecció industrial:** Detectar qualsevol imperfecció o defecte en materials fabricats, com ho és el cas d'aquest treball, identificant micro-defectes a les carrosseries de cotxes.

- **Anàlisis d'imatges per satèl·lit:** Permet obtindré informació detallada sobre la ubicació de diferents objectes al terra, amb la capacitat d'analitzar grans àrees de la terra en poc de temps.

- **Anàlisis d'imatges mèdiques:** Principalment aplicat al tractament d'imatges de ressonàncies magnètiques identificant la presència de possibles tumors elaborant diagnòstics.

- **Robòtica:** Utilitzat per realitzar tasques de forma eficient, des de tasques industrials a grans fàbriques fins als sector agrícola.

Al següent punt explicarem detalladament com funcionen aquest tipus de tècniques que estan guanyant protagonisme en tots els sectors com acabem de vorer.

3. Modelat amb U-Nets

Abans d'entrenar amb profunditat amb les xarxes neuronals amb arquitectura U-Net anem a fer una introducció a les xarxes neuronals convolucionals amb l'objectiu de comprendre millor els punts següents.

Les xarxes neuronals convolucionals són un tipus de xarxes neuronals artificials que sorgeixen com una variació del perceptró multicapa. Estan formades per un conjunt de neurones que s'assembla molt amb comportament a les neurones d'un cervell biològic. Aquest tipus de xarxes resulten molt eficients i efectives per a tasques de visió artificial, tant per a classificació com per a segmentació semàntica de imatges, entre altres possibles aplicacions.

Les xarxes neuronals convolucionals consisteixen en múltiples capes de diversos tipus i amb funcionalitats diferents, entre elles trobem:

- **Capes convolucionals**: En aquestes capes es realitza el procés d'extracció de característiques de les imatges d'entrada. Aquest procés es realitza mitjançant l'aplicació d'un filtre o nucli, que es una matriu bidimensional de pesos que al multiplicar-la per la matriu de dades d'entrada obteniu una matriu de característiques de la imatge d'entrada. Il·lustrades a la Figura 3.

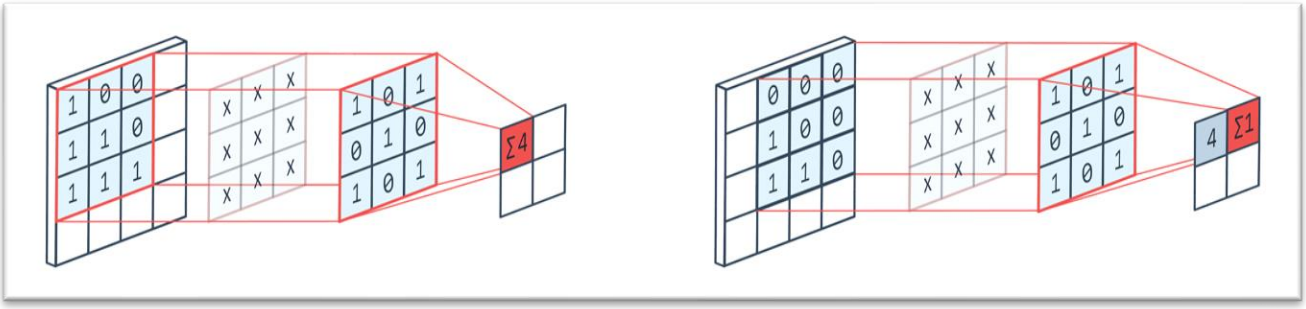


Figura 3. Il·lustració funcionament capes convolucionals

- **Funció d'activació**: Aquesta funció s'encarrega de gestionar la tassa d'aprenentatge de les neurones i quines neurones deuen activar-se en cada moment. Hi ha diverses funcions que realitzen aquesta tasca, destaquem la *ReLU*, *Leaky ReLU* i la *Sigmoid*.

- **Capes de reducció**: Aquestes capes s'encarreguen de reduir el nombre de paràmetres extrets a les capes anteriors per quedar-se sols amb les característiques més comuns o representatives. El procés consisteix en aplicar una finestra d'una dimensió definida que es mou per la matriu de característiques. Ací trobem dos tipus de capes conegudes com *MaxPooling* i *AveragePooling*. La primera d'elles agafa el valor màxim que dins de la finestra seleccionada, metre que la segona agafa una mitja de tots els valors compresos en ella. A la Figura 4 vegem un exemple d'ambdós tipus de capes.

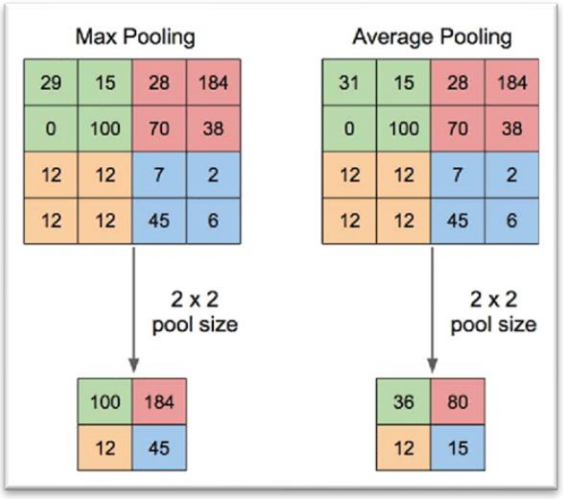


Figura 4. Il·lustració diferents capes de reducció

- **Capes de normalització:** Aquesta és una capa addicional que s'afegeix quan s'utilitzen models més complexos i profunds per tal de simplificar els vectors de característiques i obtindré així una mitjana i variància de les dades igual a cero. Trobem principalment 4 mètodes distints de normalització, *BatchNorm, LayerNorm, InstanceNorm* i *GroupNorm,* representats a la Figura 5.

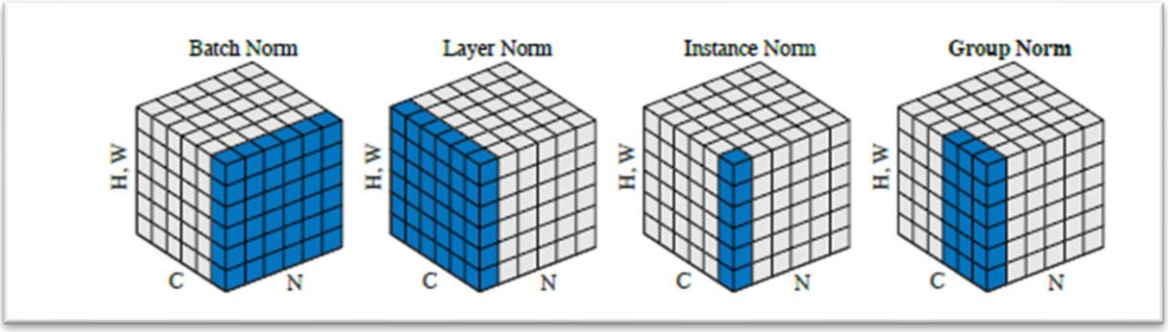


Figura 5. Il·lustració diferents capes de normalització

- **Arquitectures per classificació:** A l'última part d'una xarxa neuronal convolucional trobem la capa de classificació on s'assigna una classe a cada píxel de la imatge d'entrada a partir del vector de característiques extret a les capes anteriors. Algunes de les arquitectures més conegudes per a aquesta tasca són: *LeNet, AlexNet, ResNet i DenseNet* entre altres.

3.1 U-Net per a segmentació semàntica d'imatges

Tal i com he pogut veure a capítols anteriors, la tasca de segmentació semàntica d'imatges consisteix en assignar una classe a cada píxel de la imatge d'entrada, on les classes poden ser cotxes, carreteres, persones, etc. A diferència de la segmentació d'instàncies, tots els píxels que pertanyen a una mateixa classe tindran el mateix valor associat, es a dir, dos cotxes distints de la mateixa classe tindran assignada la mateixa classe ja que no fa diferencies entre objectes.

La forma més comú d'afrontar les tasques de segmentació semàntica és utilitzant una arquitectura de codificador - descodificador (*encoder - decoder*). La part corresponent al codificador consisteix en aplicar diversos processos de convolucions per extraure característiques de xicotets detalls de la imatge d'entrada amb una resolució reduïda, aquest procés es conegut com *downsampling.* D'altra banda, el descodificar

s'encarrega de fer el procés invers al codificador, com era esperable. En aquesta part del procés s'utilitzen processos coneguts com convolucions dilatades, que consisteixen en una expansió simètrica del mapa de característiques extret al codificador. En aquesta part també s'incrementa de nou la resolució de la imatge fins tornar a la seua resolució original d'entrada, aquest procés es coneix com *upsampling.*

A la Figura 6 vegem una representació gràfica de l'arquitectura U-Net.

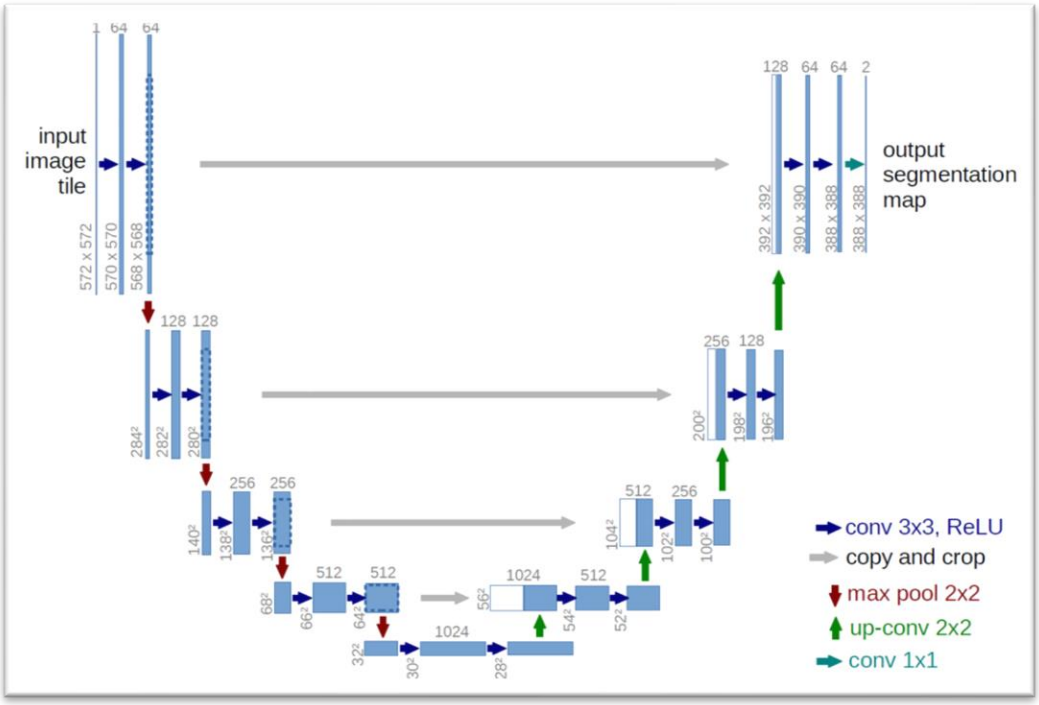


Figura 6. Disseny U-Net

3.1.1 Perquè utilitzar U-Nets

Al camp de l'aprenentatge profund es necessari tindre un gran volum de dades per entrenar les xarxes neuronals i obtenir els models esperats, però aquest és un requisit difícil de satisfer, ja siga per cost temporal, econòmic o per recursos hardware. El procés d'etiquetar les dades de les que es disposen

també requereix coneixement, experiència i dedicació.

Les U-Nets ens permeten solucionar aquests problemes, ja que han demostrat ser eficients inclús amb un conjunt de dades escàs, obtenint millor resultats que amb models convencionals. Si s'utilitzarà el que es conegut com un auto codificador clàssic, la compressió de les dades d'entrada es faria de forma lineal i impediria la transmissió de totes les característiques de la imatge. El disseny de la U-Net amb forma de U, soluciona aquest problema de pèrdua de característiques, ja que els processos de codificació i descodificació es realitzen es parts separades i es poden copiar les característiques d'una etapa a la seua equivalent del procés contrari. (Aquest procés el vegem representat a la imatge anterior amb les fletxes grises horitzontals).

Una representació d'un auto codificador clàssic la vegem a la Figura 7.

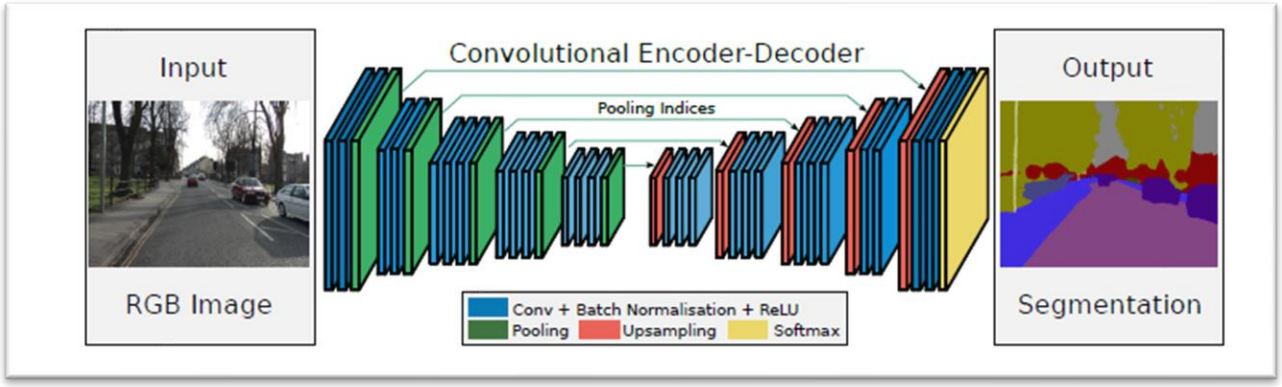


Figura 7. Il·lustració auto codificador clàssic

3.1.2 Mètriques d'avaluació

Per tal de mesurar el rendiment dels models generats pels entrenaments de les xarxes neuronals car fer ús de certes funcions que ens faciliten dades i estadístiques durant els processos d'entrenament (*train*), validació (*validation*) i test.

Aquestes mètriques es calculen a partir de dos tipus distints de funcions, les funcions de pèrdua (*loss*), i les funcions de puntuació (*score*).

Les funcions de pèrdua ens avaluen la desviació que hi ha entre les prediccions realitzades per la xarxa neuronal i les dades reals utilitzades durant l'aprenentatge. El valors estan compresos entre 0 i 1, sent 0 una predicció idèntica a la dada real i 1 tot el contrari, per tant, quant menor siga el valor de la funció de pèrdua, més eficient serà la xarxa neuronal. Per tal de minimitzar els valors de les funcions de pèrdua, es modifiquen els pesos de la xarxa després de d'analitzar cada mostra durant l'entrenament. Les funcions de pèrdua que s'han utilitzat en el desenvolupament d'aquest sistema han sigut:

A les 3 funcions el primer paràmetre que rep correspon a la dada real i el segon és la predicció realitzada per la xarxa.

- **JaccardLoss**:

  $$J(y, \tilde{y}) := \frac{|y \cap \tilde{y}|}{|y \cup \tilde{y}|}$$

- **DiceLoss**:

  $$D(y, \tilde{y}) := \frac{2|y \cap \tilde{y}|}{|y| + |\tilde{y}|}$$

- ***BCEWithLogits:*** El valor de *β* és un valor que s'ha d'ajustar per tal d'evitar obtenir falso positius. Varia depenent de les imatges d'entrada.

  $$L_{W-BCE}(y, \hat{y}) = -(\beta * ylog(\hat{y}) + (1 - y)log(1 - \hat{y}))$$

Els valors de les funcions de pèrdues que veurem als resultats dels experiments seran una mitja aritmètica dels valors obtinguts per les 3 funcions esmentades abans.

En quant a les funcions de puntuació ens mesuren de igual forma que les funcions de pèrdua quina es la desviació de la predicció de la xarxa respecte les dades reals que tenim. En el nostre cas anem a fer ús de la punció de puntuació coneguda com a *IoU score (Intersecction over Union).* Aquesta funció es pot definir com la inversa del *Jaccard loss* definit al paràgraf anterior. A la Figura 8 vegem com es realitza el càlcul i

alguns exemples d'aplicació.

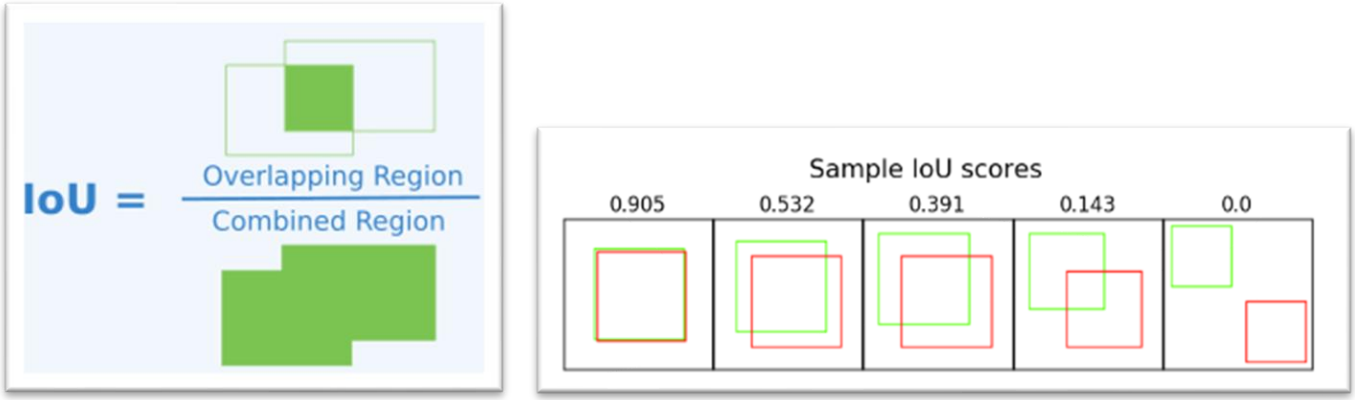


Figura 8. Il·lustració càlcul del IoU score

3.2 Arquitectures utilitzades a les U-Nets

Tal i com hem explicat abans, dins de l'estructura d'una xarxa neuronal convolucional dissenyada per a tasques de segmentació semàntica hi ha una part encarregada de classificar cada píxel de la imatge d'entrada. Al llarg dels anys s'han desenvolupat i crear noves arquitectures que realitzen aquesta classificació. A continuació nombrarem algunes de les que han tingut més rellevància i ens centrarem més amb l'escollida per al nostre cas particular.

Una de les primeres arquitectures va ser creada al 1998 per Yan LeCunn anomenada *LeNet*. La seua finalitat era classificar imatges de dígits manuscrits, entrenada amb el conjunt de dades de MNIST. El seu disseny el vegem a la Figura 9.

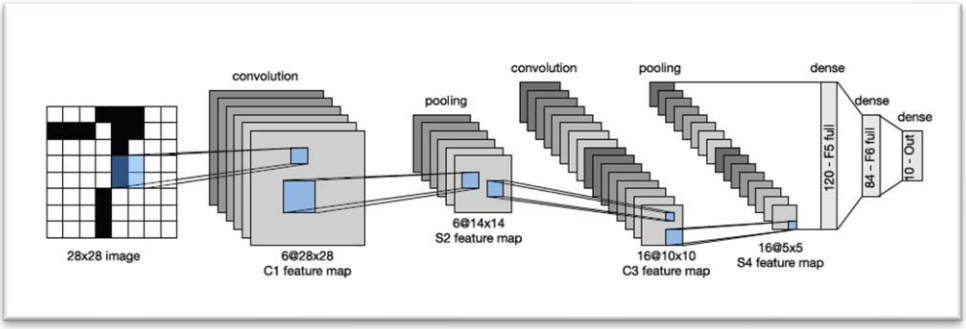


Figura 9. Il·lustració disseny AlexNet

Anys més tard, en 2012, va sorgir la AlexNet, nomenada com el seu creador, Alex Krizhevsky. Va aconseguir reduir de forma més que significativa les tasses d'error que s'obtenien amb les arquitectures anteriors. El seu disseny és molt similar la de la LeNet mostrat abans, però amb algunes diferències, és més profunda, és a dir, està formada per més capes, utilitza funcions d'activació ReLu en compte de TanH i altres detalls més.

Al 2015 va aparèixer com a guanyadora del concurs de classificació de ImageNet una estructura creada per un equip de Microsoft anomenada ResNet. Aquesta arquitectura implementa un concepte nou que no havia aparegut amb anterioritat que són els blocs residuals, d'ací el nom que rep. Aquests blocs s'encarreguen únicament de calcular la diferència entre l'entrada i l'eixida de la capa, el que és una tasca senzilla. Un bloc residual simple té la següent forma (imatge de l'esquerra):

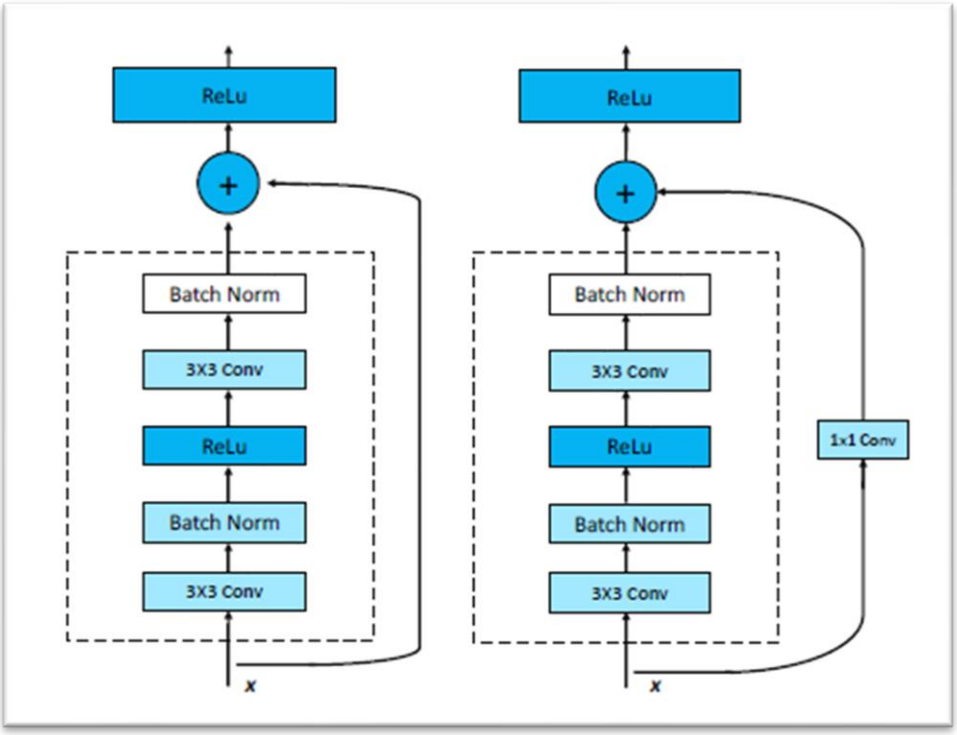


Figura 10. Bloc residual bàsic (esquerra),
Bloc residual amb connexió de bot (dreta)

Es pot assegurar que l'eixida de la capa convolucional tindrà la mateixa dimensió que l'entrada de la següent mitjançant *padding*, és a dir, afegir voreres a les imatges per obtindré la dimensió desitjada. Si també volem mantenir el nombre de canals de les imatges, cal afegir una connexió de bot d'una convolució d'1x1, com vegem a la dreta de la imatge anterior.

Aquests blocs residuals ens permeten entrenar models molt profunds, la raó és que el gradient es pot moure directament des de l'eixida fins les primeres capes mitjançant les connexions de bot.

A partir de l'arquitectura anterior que feia ús de blocs residuals, concatenant l'eixida de cada capa amb la seua entrada, si apilem diversos blocs d'aquests tipus on l'entrada de cada capa depèn de l'eixida de totes les capes que la precedeixen, tinguem el que es conegut com a *DenseNet.* De forma simple i esquemàtica es por representar com a la Figuta 11.

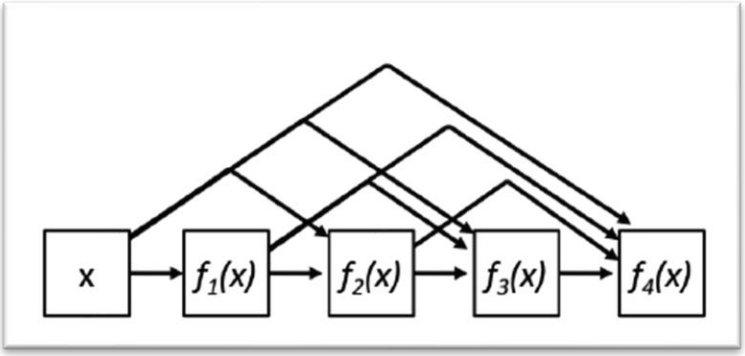


Figura 11. Il·lustració DenseNet

Més tard eixiren noves arquitectures com la *ConvNeXt, Auto-ML o EfficientNet.* No entrarem a explicar-les i passem a centrar-nos amb la *ResNet,* que es la utilitzada per al desenvolupament del sistema d'aprenentatge profund d'aquest TFG.

Dins de la família de les ResNets trobem diferents arquitectures depenent del nombre de capes que continga cadascuna d'elles. Anem a centrar-nos en 4 d'elles, la ResNet18, ResNet50, ResNet101 i ResNet152.

- ***ResNet18:*** A la Figura 12 vegem com es representaria l'esquema intern del codificador.

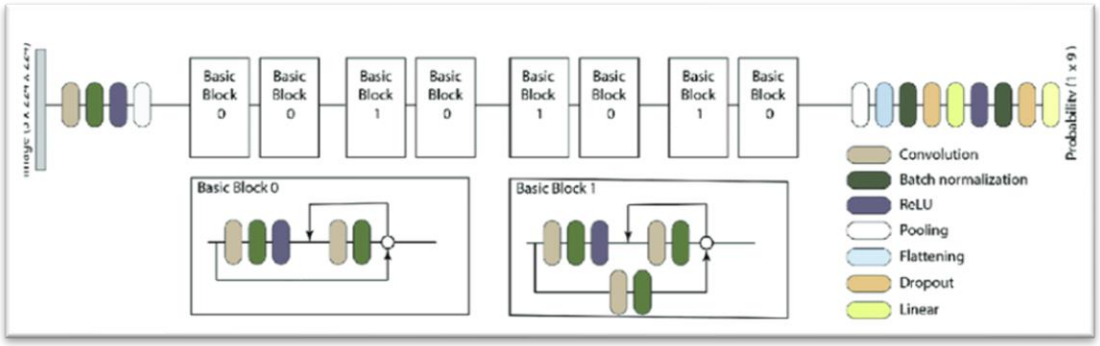


Figura 12. Esquema del codificador ResNet18.

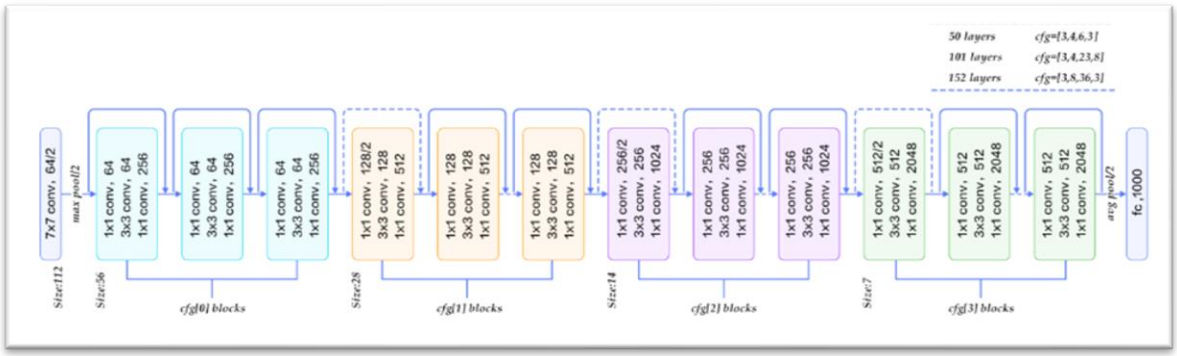


Figura 13. Esquema general per als codificadors ResNet50, ResNet101 i ResNet152.

- ***ResNet50:*** Aquesta arquitectura parteix de la base de la Figura 13, on trobem 3 instàncies del primer conjunt de blocs (blau), 4 instàncies del segon conjunt (taronja), 6 del tercer conjunt (morat) i 3 de l'últim (verd). En total formada per 50 capes.

- ***ResNet101:*** Aquesta arquitectura parteix de la base de la Figura 13, on trobem 3 instàncies del

primer conjunt de blocs (blau), 4 instàncies del segon conjunt (taronja), 23 del tercer conjunt (morat) i 8 de l'últim (verd). En total formada per 101 capes.

- ***ResNet152:*** Aquesta arquitectura parteix de la base de la Figura 13, on trobem 3 instàncies del primer conjunt de blocs (blau), 8 instàncies del segon conjunt (taronja), 36 del tercer conjunt (morat) i 3 de l'últim (verd). En total formada per 152 capes.

3.3 Experiments amb U-Nets

A aquest punt anem a detallar els diversos experiments que s'han realitzat amb les xarxes neuronals convolucionals amb arquitectura U-Net explicades als punts anteriors.

El primer dels experiments serà el realitzar amb el primer dels conjunts de dades detallats al punt 2.2 d'aquesta memòria, on les imatges seran de dimensió 320x256 píxels.

Per aquest i per a la resta d'experiments que s'exposaran a la memòria s'han fet 4 proves, cadascuna amb un dels *encoders* de *resnet* explicats al punt anterior.

Comencem pel primer d'ells:

Les gràfiques resum de cadascun dels entrenaments les podem trobar a l'Annex. En aquestes gràfiques veurem a l'esquerra la gràfica amb la representació dels valors mitjans de les funcions de pèrdua en cada *epoch,* i a la dreta la gràfica amb la representació dels valors mitjans de la funció de puntuació IoU de la cada *epoch.*

Amb la finalitat d'analitzar de forma més còmoda i clara representem tots els resultats dels entrenaments realitzats amb dues taules comparatives.

Taula 1. Comparativa resultats funcions de pèrdua i temps d'entrenament

Encoder	Train Loss	Val Loss	Test Loss	Train Time (h)
resnet18	0.1747	0.1739	0.1754	1.12 (4020)
resnet50	0.1749	0.1738	0.1753	1.82 (6568)
resnet101	0.1747	0.1739	0.1754	2.45 (8803)
resnet152	0.1748	0.1739	0.1756	4.18 (15045)

Taula 2. Comparativa resultats de funció de puntuació IoU d'entrenament, validació i test.

Encoder	Train IoU	Val IoU	Test IoU
resnet18	98.93%	99.23%	99.08% (±0.58%)
resnet50	98.90%	99.25%	99.09% (±0.58%)
resnet101	98.94%	99.24%	99.09% (±0.60%)
resnet152	98.88%	98.87%	99.05% (±0.61%)

Si observem els resultats de la Taula 1, vegem que les diferències entre els 4 *encoders* apareixen al quart decimal, pel que no ens aporta cap informació sobre quin dels 4 es l'òptim. Per això la columna amb el temps emprat a completar l'entrenament és la que ens dona informació més clarificadora, ja que el *resnet18* gaire bé tarda 1h en completar l'entrenament mentre que el *resnet152* sobrepassa les 4h. Es completament lògic, ja que el *resnet18* es el que menys paràmetres calcula i el *resnet152* el que més de tots 4.

La Taula 2 amb els valors de puntuació IoU no ens aporta cap diferència significativa entre les diferents proves, ja que totes 4 obtenen un 99% d'encert en les proves de test.

Per tant, amb aquesta informació podem concloure que l'opció òptima és utilitzar el *resnet18* ja que obté uns resultats pràcticament idèntics a la resta però amb una reducció notable en el temps d'entrenament.

3.4 Variants factor d'escala

Vists els resultats d'aquest primer experiment es plantegen dos experiments addicionals per comparar resultats i veure amb quin s'aconsegueixen millors resultats. Aquests experiments addicionals consisteixen en modificar la resolució de les imatges amb les quals es realitzen tant l'entrenament, la validació com el test. També s'ha modificat el conjunt de dades utilitzat, per a aquests experiments i els posteriors es farà servir el segon conjunt de dades esmentat a l'apartat 2.2 de la memòria. D'aquest conjunt dades extraguem dues variants, una amb resolució de 640x512 píxels i una altra més menuda de 160x128 píxels, com es pot veure sempre guardant la mateixa relació d'aspecte. Com a recordatori, aquest *dataset* està format tant per imatges de Volkswagen com de Mercedes-Benz.

De igual forma que als experiments de l'apartat anterior, es realitzen 4 proves amb els mateixos 4 *encoders* distints explicats anteriorment.

Comencem primer amb l'experiment amb imatges de resolució xicoteta, **160x128 píxels.**

Les gràfiques resum de cadascun dels entrenaments les podem trobar a l'Annex. En aquestes gràfiques veurem a l'esquerra la gràfica amb la representació dels valors mitjans de les funcions de pèrdua en cada *epoch,* i a la dreta la gràfica amb la representació dels valors mitjans de la funció de puntuació IoU de la cada *epoch.*

Amb la finalitat d'analitzar de forma més còmoda i clara representem tots els resultats dels entrenaments realitzats amb dues taules comparatives.

Taula 3. Comparativa resultats funcions de pèrdua i temps d'entrenament

Encoder	Train Loss	Val Loss	Test Loss	Train Time (h)
resnet18(bs:256)	0.1869	0.1908	0.1873	0.87 (3138)
resnet50(bs:128)	0.1814	0.1829	0.1817	1.20 (4322)
resnet101(bs:64)	0.1806	0.1809	0.1812	1.27 (4580)
resnet152(bs:64)	0.1799	0.1801	0.1803	1.45 (5227)

Taula 4. Comparativa resultats de funció de puntuació IoU d'entrenament, validació i test.

Encoder	Train IoU	Val IoU	Test IoU

resnet18(bs:256)	98.77%	100%	96.58% (±2.14%)
resnet50(bs:128)	98.54%	99.99%	96.59% (±1.80%)
resnet101(bs:64)	98.36%	99.12%	96.50% (±1.80%)
resnet152(bs:64)	98.39%	99.17%	96.55% (±1.83%)

Observant els resultats de la Taula 3 vegem una xicoteta millora amb el *resnet152* respecte als altres 3 *encoders.* Es veritat que el temps emprat per a realitzar l'entrenament és lleugerament superior a la resta, però eixe temps extra ens aporta millors resultats, pel que és la millor opció en un principi.

L'objectiu que es buscava amb aquest experiment era augmentar el *batch size* al màxim per veure si notàvem alguna diferència significativa als valors obtinguts a les funcions de puntuació IoU en l'etapa de *test.* A la Taula 4 vegem plasmats aquests resultats, amb valors molt similars entre ells, sense cap diferència significativa.

Si comparem amb la taula anterior vegem que els valor són inferiors als obtinguts a l'experiment anterior, açò es deu que al reduir tant la imatge original es perden detalls que afecte al rendiment de la xarxa neuronal. Un altre motiu que afecta en menor mesura es que ara la xarxa també té informació de carrosseries d'una factoria diferent i no sols d'una com passava al cas anterior.

Passem ara a l'altre experiment esmentat abans, amb les imatges de resolució gran, **640x512 píxels.**

Amb la finalitat d'analitzar de forma més còmoda i clara representem tots els resultats dels entrenaments realitzats amb dues taules comparatives.

Taula 5. Comparativa resultats funcions de pèrdua i temps d'entrenament

Encoder	Train Loss	Val Loss	Test Loss	Train Time (h)
resnet18	0.1769	0.1767	0.1744	27.92 (100505)
resnet50	0.1774	0.1773	0.1749	30.03 (129728)
resnet101	0.1776	0.1775	0.1749	41.90 (150864)
resnet152	0.1723	0.1722	0.1723	13.56 (48802)**

Taula 6. Comparativa resultats de funció de puntuació IoU d'entrenament, validació i test.

Encoder	Train IoU	Val IoU	Test IoU

resnet18	98.70%	97.69%	97.21% (±1.16%)
resnet50	98.64%	98.55%	97.12% (±1.36%)
resnet101	98.62%	98.52%	97.11% (±1.27%)
resnet152	99.39%	99.36%	98.74% (±2.08%)

***Aquest entrenament s'ha fet amb un equip amb major potència gràfica que els altres.*

Al cas de les imatges de resolució menuda havien dit que l'objectiu era augmentar el màxim possible el *batch size* per veure si la xarxa era capaç d'aprendre millor les característiques de les imatges d'entrada. En aquest cas el que fem es el contrari, reduir el *batch size* al mínim per entrenar amb les imatges més grans suportades per la targeta gràfica. L'objectiu es que al reduir en menor mesura les imatges que els casos anteriors, conservar un major nivell de detall, aportant major informació a la xarxa i per tant obtenir resultats de major precisió.

El resultats de la Taula 5 ens deixen entre veure un increment substancial del temps total d'entrenament, com a resultat dels motius exposats al paràgraf anterior. En quant als valors de les funcions de pèrdua podem avançar que l'*encoder resnet152* té valors inferiors a la resta i és probable que aconsegueixi millors resultats en la funció de puntuació IoU.

Si comparem a partir dels resultats de la Taula 4 i Taula 6 els millors resultats obtinguts en cadascun dels dos experiments vegem que a la prova amb imatges menudes hem obtingut una precisió del 96.50% aproximadament amb l'*encoder resnet50,* i a la prova amb imatges grans hem obtingut un 98.74% aproximadament amb el *resnet152,* ratificant el que avançàvem analitzant la Taula 5. Com poder apreciar, a pesar de les oscil·lacions que puguem sofrir aquests valors per les desviacions típiques calculades en cada cas, vegem que les xarxes de entrenades amb les imatges de major resolució obtenen resultats significativament millors que les imatges menudes.

4. Data Augmentation

Un des problemes més comuns a les tasques de visió artificial i segmentació semàntica és l'escassesa de dades disponibles per fer entrenaments grans per nodrir de la major quantitat d'informació possible a la xarxa. Una de les tècniques més aplicades al sector per solucionar aquests problemes és aplicar tècniques d'augment de dades (*data aumentation*).

Aquestes tècniques es basen en generar de forma artificial mitjançant una sèrie de transformacions noves dades a partir de les que ja es disposa. D'aquesta forma estem enriquint el nostre conjunt de dades sense necessitat d'adquirir noves dades reals, que sol ser una tasca complicada com ja hem comentat en anterioritat.

Cal anar en compte quan s'apliquen les transformacions, ja que es pot donar el cas que les imatges pateixen unes deformacions que les allunye del que serien les dades reals. Per això cal estudiar quines transformacions són adequades per a cada casuística possible.

Per al nostre cas hem aplicat les següents transformacions:

- Rotacions i inclinacions de diferents angles.
- Afegir soroll, però molt lleu per no distorsionar en excés la imatge.
- Canvis en el contrast de la il·luminació.
- Canvis de saturació dels colors.
- Efectes de difuminat.

Per poder aplicar de forma fàcil i ràpida aquestes transformacions a cadascuna de les imatges d'entrada s'ha emprat una llibreria de *Python* dissenyada per a aquesta tasca, anomenada *albumentations.* A aquesta llibreria trobem infinitat de transformacions possibles que podem a aplicar a les imatges del nostre conjunt de dades, de totes elles hem fet una selecció de les que encaixen millor amb les situacions que es poden donar a casos reals.

La finalitat d'aplicar aquesta tècnica d'augment de dades és veure si som capaços de millorar els resultats obtinguts fins al moment als experiments previs. Per tant, anem a repetir els 3 experiments grans realitzats a l'apartat 3 però ara amb la diferència d'aplicar les transformacions esmentades al conjunt de dades d'entrenament.

Per seguir amb el mateix ordre en el que hem presentat els experiments anteriors, anem a començar primerament amb les imatges del conjunt de dades exclusiu de Volkswagen amb una resolució de 320x256 píxels, seguit de les imatges del conjunt de dades format per imatges de les dues factories,

Mercedes-Benz i Volkswagen, amb les seues dues variants, les de resolució 160x128 píxels i les de 640x512 píxels.

➢ 320x256 – 1er Conjunt de dades

Les gràfiques resum de cadascun dels entrenaments les podem trobar a l'Annex. En aquestes gràfiques veurem a l'esquerra la gràfica amb la representació dels valors mitjans de les funcions de pèrdua en cada *epoch,* i a la dreta la gràfica amb la representació dels valors mitjans de la funció de puntuació IoU de la cada *epoch.*

Amb la finalitat d'analitzar de forma més còmoda i clara representem tots els resultats dels entrenaments realitzats amb dues taules comparatives.

Taula 7. Comparativa resultats funcions de pèrdua i temps d'entrenament

Encoder	Train Loss	Val Loss	Test Loss	Train Time (h)
resnet18	0.1880	0.1810	0.1838	1.82 (6534)
resnet50	0.1885	0.1815	0.1838	Sense dades
resnet101	0.1890	0.1815	0.1839	Sense dades
resnet152	0.1890	0.1825	0.1847	5.58 (20100)

Taula 8. Comparativa resultats de funció de puntuació IoU d'entrenament, validació i test.

Encoder	Train IoU	Val IoU	Test IoU
resnet18	97.70%	98.15%	98.03% (±1.04%)
resnet50	97.75%	98.00%	97.99% (±1.14%)
resnet101	97.60%	98.00%	97.97% (±1.22%)
resnet152	97.60%	97.40%	97.87% (±1.34%)

Si comparem els resultats de les Taules 7 i 8 amb els obtinguts a l'experiment homogeni a aquest però sense *data aumentation* (Taules 1 i 2), vegem que aquests no els milloren en cap de les proves. Per tant desestimarem aquest sistema donat que l'anterior tenia millor rendiment.

➢ 160x128 – 2on Conjunt de dades

Les gràfiques resum de cadascun dels entrenaments les podem trobar a l'Annex. En aquestes gràfiques veurem a l'esquerra la gràfica amb la representació dels valors mitjans de les funcions de pèrdua en cada *epoch,* i a la dreta la gràfica amb la representació dels valors mitjans de la funció de puntuació IoU de la cada *epoch.*

Amb la finalitat d'analitzar de forma més còmoda i clara representem tots els resultats dels entrenaments realitzats amb dues taules comparatives.

Taula 9. Comparativa resultats funcions de pèrdua i temps d'entrenament

Encoder	Train Loss	Val Loss	Test Loss	Train Time (h)
resnet18	0.2026	0.2012	0.1977	1.87 (6754)
resnet50	0.1955	0.1923	0.1914	2.25 (8090)
resnet101	0.1947	0.1906	0.1910	1.76 (6345)
resnet152	0.1944	0.1898	0.1901	2.02 (7286)

Taula 10. Comparativa resultats de funció de puntuació IoU d'entrenament, validació i test.

Encoder	Train IoU	Val IoU	Test IoU
resnet18	96.78%	99.82%	95.29%(±2.57%)
resnet50	96.85%	98.41%	95.46%(±2.53%)
resnet101	96.58%	97.44%	95.37%(±2.82%)
resnet152	96.61%	97.56%	95.57%(±2.64%)

Comparant les Taules 3 i 9, i les Taules 4 i 10 respectivament, vegem un empitjorament d'un 1% aproximadament. Comparem aquestes taules ja que corresponen als experiments homogenis, sense i amb *data augmentation* respectivament. Per tant, podem concloure que l'experiment homogeni anterior té una major precisió i descartem aquest últim.

➢ 640x512 – 2on Conjunt de dades

Taula 11. Comparativa resultats funcions de pèrdua i temps d'entrenament

Encoder	Train Loss	Val Loss	Test Loss	Train Time (s)
resnet18(bs:256)	0.1868	0.1877	0.1850	53.11 (191202)
resnet50(bs:128)	0.1881	0.1884	0.1864	60.83 (218974)
resnet101(bs:64)	0.1906	0.1877	0.1886	9.81 (35318)**
resnet152(bs:64)	0.1909	0.1880	0.1884	10.85 (46271)**

***Aquest entrenament s'ha fet amb un equip amb major potència gràfica que els altres.*

Taula 12. Comparativa resultats de funció de puntuació IoU d'entrenament, validació i test.

Encoder	Train IoU	Val IoU	Test IoU
resnet18(bs:256)	97.69%	96.79%	95.94% (±2.33%)
resnet50(bs:128)	97.47%	96.68%	95.73% (±2.65%)
resnet101(bs:64)	96.94%	96.70%	97.01% (±3.09%)
resnet152(bs:64)	96.90%	96.66%	96.94% (±3.03%)

De igual forma que als 2 experiments previs amb *data augmentation,* no vegem cap milloria als resultats. Comparant les Taules 5 i 6 amb les Taules 11 i 12 apreciem més bé un empitjorament dels resultats anteriors al voltant d'un 1%. Aquest sistema també es descarta perquè el seu homogeni sense l'aplicació d'aquesta tècnica obté millors resultats.

5. Entrenament amb patches

En aquest capítol anem a descriure l'últim experiment realitzat per tal de millorar els resultats obtinguts als experiments previs. Com ara passarem a explicar, aquesta tècnica d'entrenament amb *patches* (pegats), és una prova que requereix un alt consum de recursos i un temps prolongat d'entrenament, per això farem una única prova, que serà al millor dels models obtinguts a les proves realitzades anteriorment.

És cert que hem obtingut una precisió del 99% amb els experiments realitzats amb el conjunt de dades de Volkswagen, però creguem que els resultats amb un 98.74% de precisió amb les imatges grans (640x512 píxels) amb l'*encoder resnet152* és millor ja que conté informació de dues factories i per tant té una millor capacitat de generalització de cara a possibles noves carrosseries.

Aquesta tècnica consisteix en partir cada imatge d'entrada en *patches* (pegats), de forma que la imatge que rep la xarxa com a entrada és un xicotet fragment de la imatge completa, de forma que facilita la tasca per tal d'obtenir més informació del detalls més menuts. Com hem dit abans aquesta tècnica requereix més recursos i temps d'entrenament, ja que per a cada imatge extrau molts fragments de la mateixa que la xarxa ha de processar. Aquest cost extra es deu traduir amb un millor rendiment del sistema ja que està processant moltes imatges i amb més detall que amb un entrenament convencional.

Per poder dur a terme aquest experiment i obtenir resultats s'ha hagut de reduir el nombre de dades dels conjunts d'entrenament i validació, utilitzant menys del 10% de les dades totals. Concretament hem utilitzat 2000 imatges d'entrenament i 350 de validació, en ambdós conjunts les imatges són 50% de Volkswagen i 50% de Mercedes-Benz. Aquesta modificació ha sigut forçada pels motius explicats als paràgrafs anteriors on s'explicava la seua complexitat computacional i la necessitat de grans màquines per poder dur-los a terme en condicions òptimes.

A la Figura 14 podem veure les gràfiques resultat de l'entrenament. Com podem observar les corbes que representen els valors de les etapes d'entrenament i validació estan prou separades entre elles, i no arriba a obtenir bons resultats en validació. Açò es pot deure a la escasses de les dades proporcionades per a l'entrenament, ja que hem un nombre tant xicotet la xarxa no es capaç d'aprendre tots els detalls necessaris per arribar als nivells de precisió desitjats.

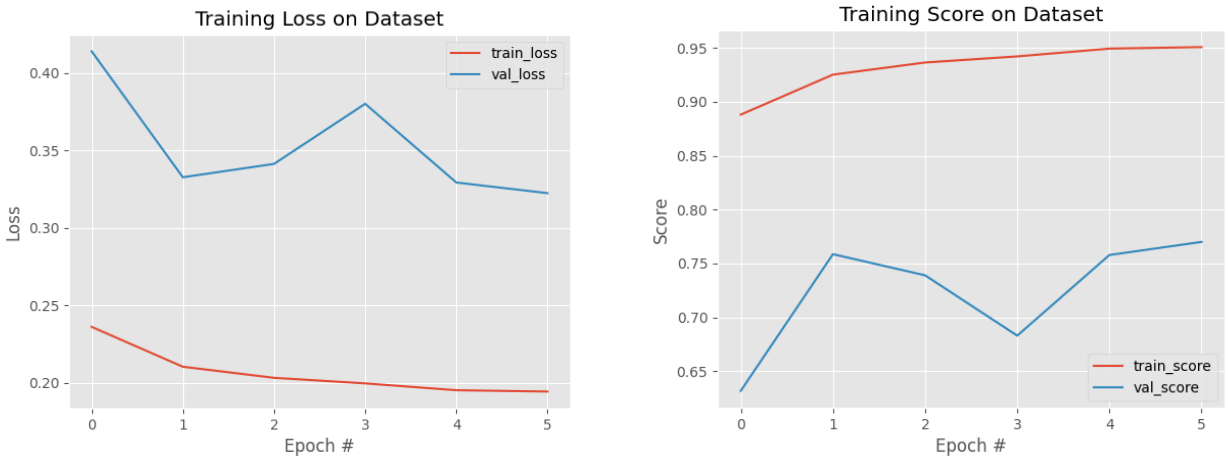


Figura 14. A l'esquerra gràfica funcions de pèrdua, a la dreta gràfica funció de puntuació IoU.

Per visualitzar els resultats més fàcilment proporcionem la següent taula:

Taula 13. Comparativa resultats funcions de pèrdua i temps d'entrenament

Encoder	Train Loss	Val Loss	Test Loss	Train Time (s)
resnet152	0.1942	0.3223	0.3484	56890

Taula 14. Comparativa resultats de funció de puntuació IoU d'entrenament, validació i test.

Encoder	Train IoU	Val IoU	Test IoU
resnet152	95.07%	77.06%	80.59% (±6.08%)

A les Taules 13 i 14 corroborem numèricament el que hem comentat abans, la xarxa necessita major nombre de dades per millorar el seu rendiment en les etapes de validació i test.

6. Conclusions

Després d'analitzar i explicar detalladament tots els entrenaments i proves realitzades amb les xarxes neuronals profundes amb arquitectura U-Net, podem extraure diverses conclusions sobre quin és el seu funcionament. Al present TFG s'ha enfocat molt l'atenció en una aplicació concreta, como ho és la tasca de segmentació semàntica d'imatges, en un dels casos més simples, ja que sols hi havia que identificar una sola classe a part del fons, que era la zona d'inspecció de la carrosseria.

Tornant als objectius plantejats a l'inici de la memòria, recordem que l'objectiu bàsic era crear d'un sistema d'aprenentatge profund per a la generació automàtica de màscares de detecció que transforme un procés manual, costós i repetitiu en un automàtic i més eficient. S'ha pogut dur a terme la creació i les proves pertinents d'aquest sistema de forma satisfactòria, acomplint amb els nivells de precisió que requeria un sistema d'aquestes característiques. S'ha comprovat que la creació d'un sistema automàtic dins d'un procés industrial redueix molt notablement el temps que comportava realitzar eixa tasca a un tècnic. El fet de poder alliberar a un treballador de fer una tasca repetitiva que ara serà realitzada de forma automàtica genera recursos disponibles, en aquest cas a la factoria, que pot re-assignar a un altre punt de la cadena de producció per incrementar la eficiència, i a llarg termini convertir-se en benefici econòmic. Per tant, es pot afirmar que s'ha complit l'objectiu bàsic d'aquest treball.

Seguint la seqüència d'objectius principals plantejats per acomplir l'objectiu bàsic, s'han extret les següents conclusions en el transcurs del treball realitzat:

1. S'han ampliat els coneixements sobre les principals tasques que poden realitzar les xarxes neuronals amb imatges, s'han explicat en que consisteixen cadascuna d'elles i en que es diferencien. Amb tota la informació s'ha elegit aquella que s'adaptava i que podria proporcionar una solució al problema inicial, generant màscares de forma automàtica.

2. S'ha profunditzat en els sistema de xarxes neuronals profundes per a realitzar tasques de segmentació semàntica d'imatges i les seues possibles aplicacions en altres àmbits o sectors. S'ha pogut veure numèricament com afecta la resolució de les imatges d'entrada a la xarxa al seu rendiment. Es veu una clara milloria als models entrenats amb les imatges de major resolució respecte a les de menor. Amb aquests resultats cal pensar que amb equips o *clústers* amb major

potencia computacional, el rendiment dels sistemes com el creat en aquest TFG, podria ser molt elevat ja que seria possible treballar amb dades d'entrada de resolució completa sense perdre cap detall.

3. A partir dels experiments realitzats amb dades reals hem pogut observar la milloria en ¡ la qualitat de les màscares generades en quant a l'ajust exacte a cada imatge d'entrada. Prèviament es generava únicament una màscara de cada càmera de cada model, i després eixa màscara s'ajustava a cada imatge d'entrada mitjançant algorismes de *matching,* corregint els possibles desplaçaments o obertures. Amb el nou sistema automàtic s'elimina la necessitat d'aquests algorismes de correcció ja que les màscares es generen directament a partir de la imatge d'entrada i s'ajusten a ella.

6.1 Ampliacions

Durant la realització del treball i el seu corresponent procés d'investigació, es veuen infinitat d'aplicacions i idees per treballar amb les xarxes neuronals que fan reflexionar i ajuden a millorar el teu propi projecte. D'esta forma s'han recopilat el que es consideren les principals millores o camins a seguir per tal d'optimitzar el rendiment del sistema creat. Algunes d'elles són:

- Realitzar entrenaments amb imatges de resolució completa, és a dir, de 2592x2048 píxels. Com hem dit abans, a major resolució d'imatge, més característiques d'ella pot aprendre la xarxa. Per tal de poder dur a terme aquests experiments cal disposar d'un *clúster* amb gran potència computacional per tractar imatges tant pesades com aquestes.

- Relacionat amb el punt anterior, caldria fer entrenament més prolongats en el temps, per veure fins on pot aprendre la xarxa sense arribar al sobre entrenament. Un exemple podria ser entrenar amb 50, 100, 150 i 200 *epochs* i analitzar els resultats i veure si la milloria és notable.

- Repetir l'experiment amb *patches* del punt 5 de la memòria amb un equip que permet processar el conjunt de dades al complet i durant major nombre d'iteracions.

- Seguint amb les imatges de resolució completa, fer un entrenament amb *patches*. Provar a extraure el número màxim de *patches* amb solapament de la imatge per obtenir més informació de detalls com puguin ser manetes de les portes, retrovisors, etc.

- Com a millora d'eficiència de la proposta anterior, es podrien extraure *patches* únicament de les zones on estan els detalls importants, és a dir, no són zones planes de la carrosseria com ho pot ser la part central d'una porta o del sostre. Aquest millora afegeix major complexitat al problema ja que hi ha que fer un processat de la imatge abans de ser tractada per la xarxa.

6.2 Relació del treball amb els estudis cursats

El desenvolupament del sistema automàtic per a la generació de màscares de zones d'interès a carrosseries de cotxes, ha servit per posar en valor tots els coneixements apresos des de l'inici del Grau en Enginyeria Informàtica.

Primerament, tots els conceptes bàsics sobre la programació apresos al llarg de la carrera, des de les bases de la programació, el domini de les diferents estructures de dades, fins treballar amb sistemes complexos com ho són les xarxes neuronals.

Les metodologies de la enginyeria software estudiades i aplicades per dissenyar el codi de forma eficaç i estructurada, de igual forma que identificar correctament les diferents fases que conformen el desenvolupament del treball.

Els conceptes apresos a les assignatures de Sistemes Intel·ligents, Percepció i Aprenentatge Automàtic han sigut de gran ajuda per poder entendre el funcionament al més baix nivell de les xarxes neuronals. També aprendre la metodologia i passos a seguir per a realitzar experiments de forma correcta i com obtindré i representar la informació més rellevant dels mateixos.

La gestió de projectes per saber gestionar els recursos d'un projecte, organitzar el treball a realitzar en

tasques més menudes i distribuir-les entre un equip de treball. També la forma d'avaluar setmanalment els avanços i problemes que poden anar sorgint al llarg del desenvolupament.

Per tots aquests coneixements i altres apresos en altres rames de la informàtica que no s'esmenten en aquets TFG, es considera que els estudis del Grau en Enginyeria Informàtica han sigut satisfactoris i tot el treballar realitzat durant la carrera han fet possible realitzar un projecte com aquest.

7. Bibliografia

1.

Murphy KP. [Internet]. Probabilistic Machine Learning: An Introduction. 2022.
Disponible en: https://probml.github.io/pml-book/book1.html

2.

Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL. SEMANTIC IMAGE SEGMENTATION WITH DEEP CONVOLUTIONAL NETS AND FULLY CONNECTED CRFS [Internet]. 2016.

Disponible en: https://arxiv.org/pdf/1412.7062.pdf

3.

Posada R. USING DEEP LEARNING TECHNIQUES FOR SEMANTIC SEGMENTATION OF IMAGES [Internet]. 2021.
Disponible en:
https://www.google.es/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&ved=2ahUKEwjW9uOfo6f2AhVNr6QKHVzyAPgQFnoECDgQAQ&url=https%3A%2F%2Fupcommons.upc.edu%2Fbitstream%2Fhandle%2F2117%2F345282%2FDegree_thesis_Roger_Posada.V3.pdf%3Fsequence%3D3&usg=AOvVaw2aDgh9-StBCABHMV6Fgw0M

4.

Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation [Internet]. 2015.
Disponible en: https://link.springer.com/content/pdf/10.1007%2F978-3-319-24574-4_28.pdf

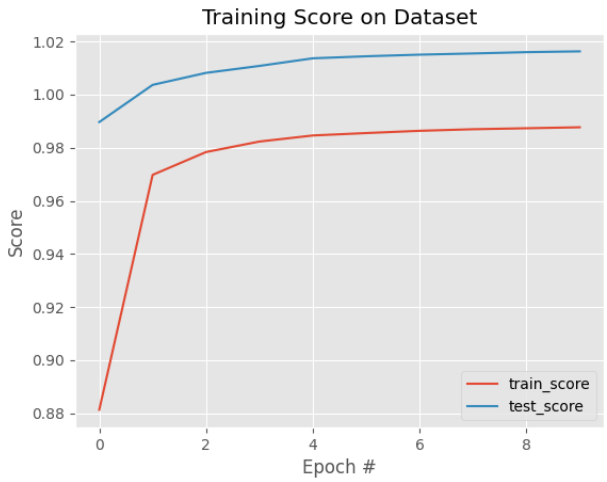
5.

Tiu E. Metrics to Evaluate your Semantic Segmentation Model [Internet]. 2019 [citado 18 mayo 2022]. Disponible en: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

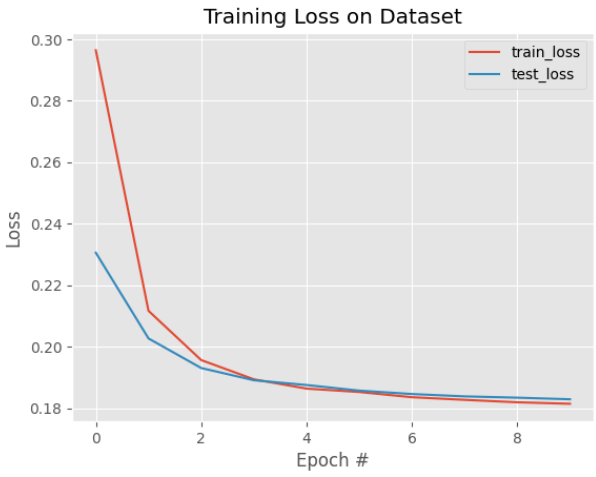
Annex I – Gràfiques entrenaments 160x128 píxels

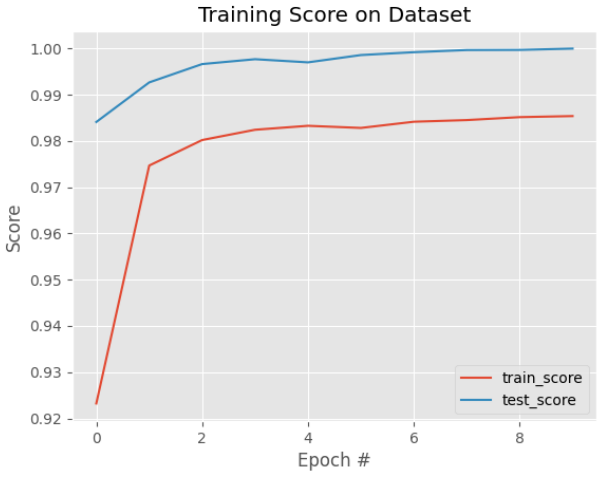
➢ Experiments sense data augmentation

❖ *Resnet18:*



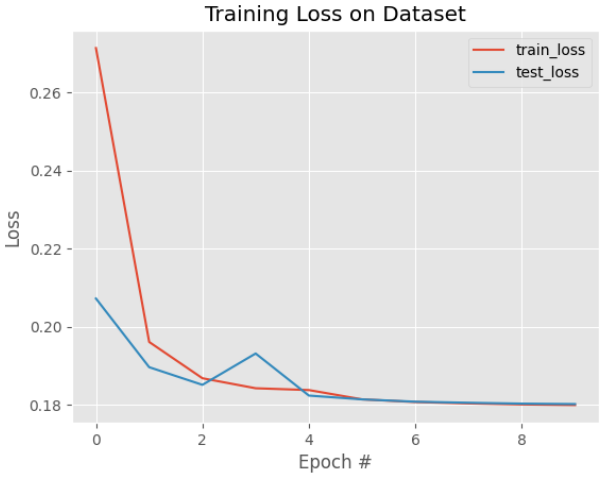
❖ *Resnet50:*

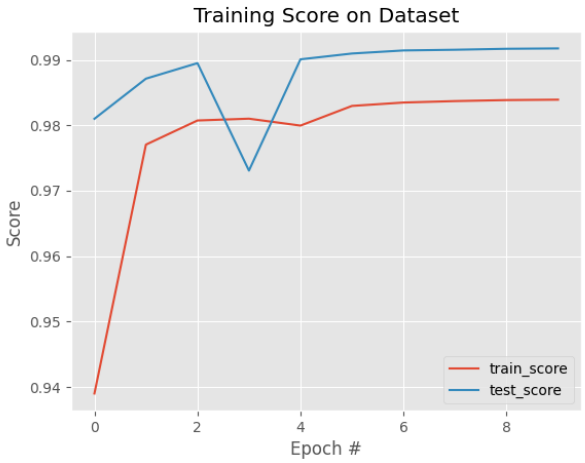




❖ ***Resnet101:***

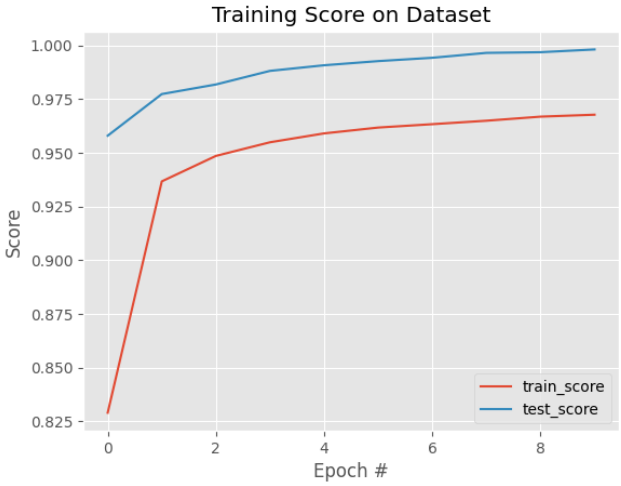
❖ ***Resnet152:***





- **Experiments amb data augmentation:**

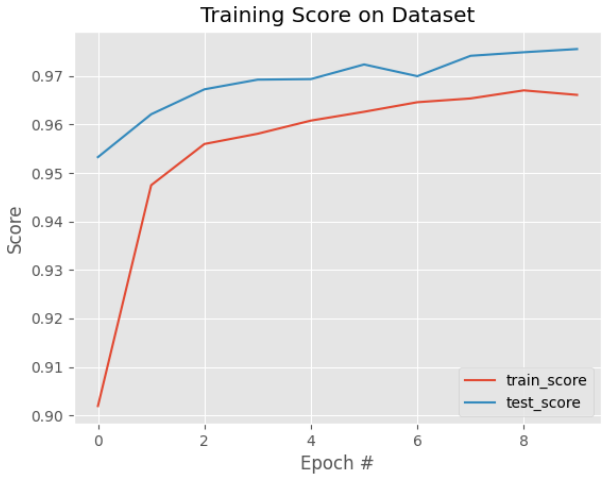
  - ***Resnet18:***



  - ***Resnet50:***

❖ ***Resnet101:***

❖ ***Resnet152:***

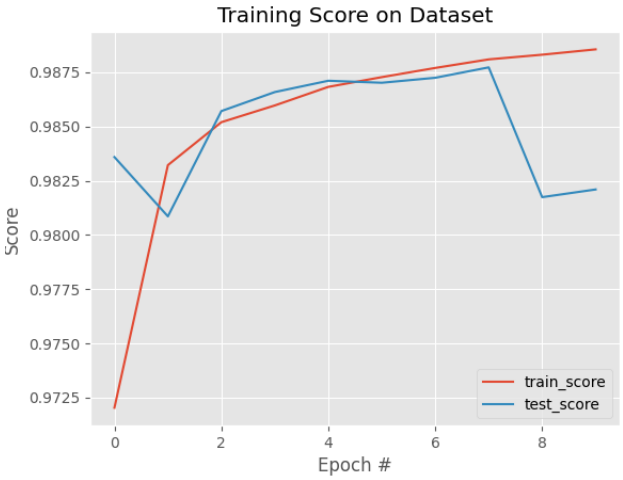


Annex II – Gràfiques entrenaments 320x256 píxels

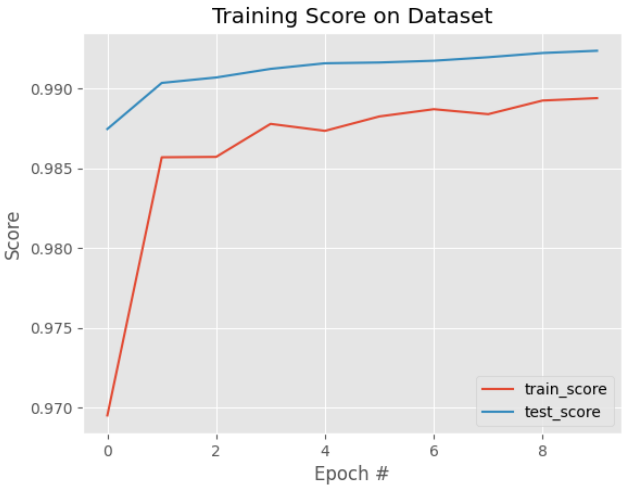
➢ **Experiments sense data augmentation**

❖ ***Resnet18:***

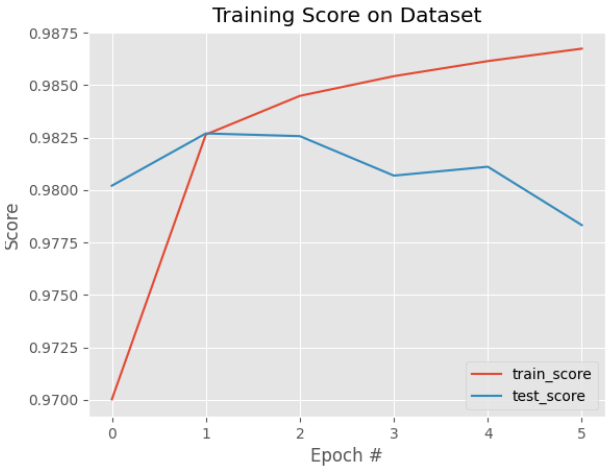
❖ ***Resnet50:***



❖ ***Resnet101:***



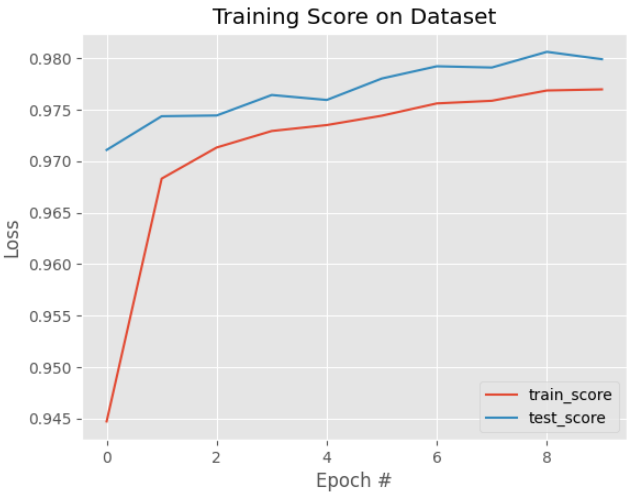
❖ ***Resnet152:***



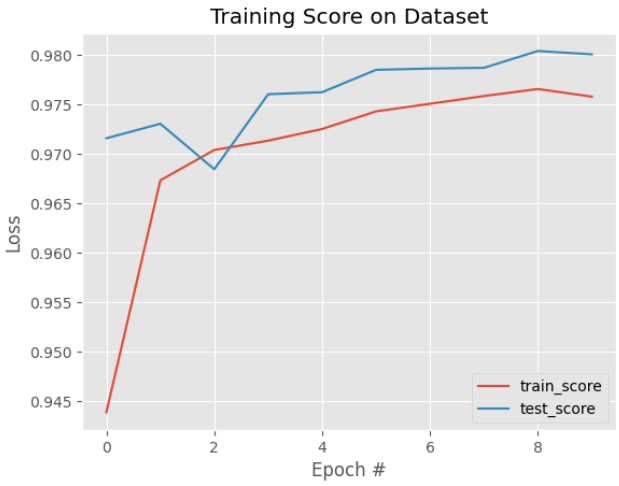
- **Experiments amb data augmentation:**

  - ***Resnet18:***

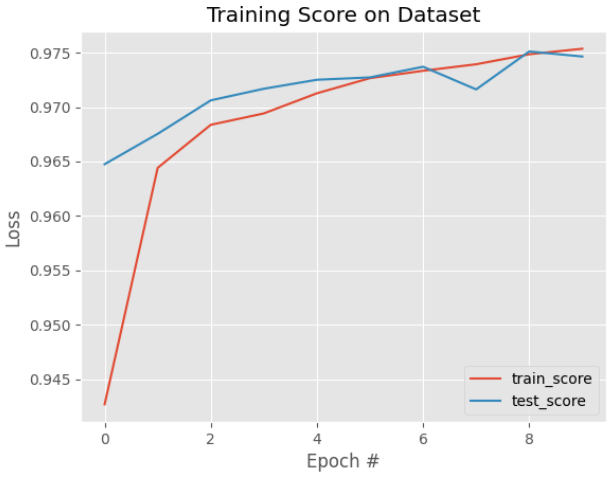
  - ***Resnet50:***



❖ ***Resnet101:***



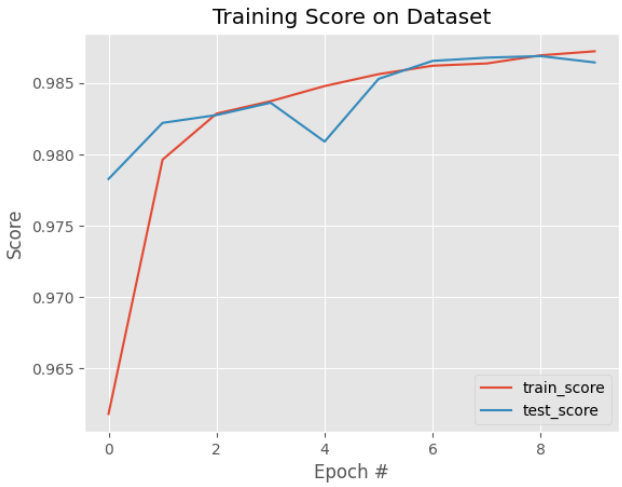
❖ ***Resnet152:***



Annex III – Gràfiques entrenaments 640x512 píxels

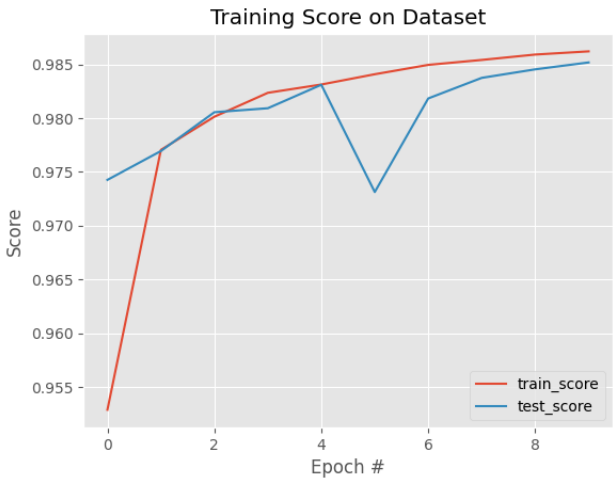
➢ **Experiments sense data augmentation**

❖ ***Resnet18:***

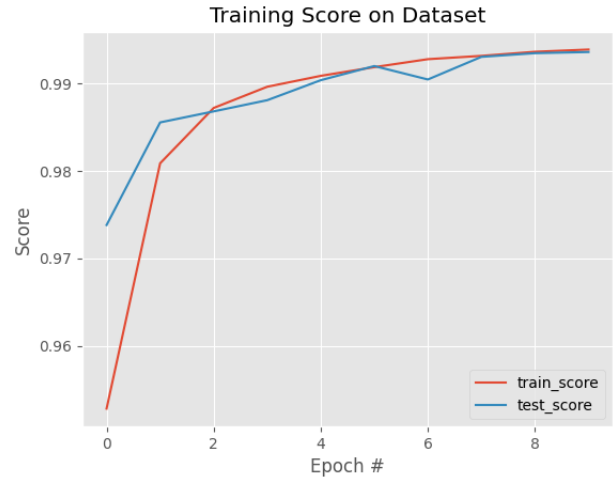


❖ ***Resnet50:***

❖ ***Resnet101:***



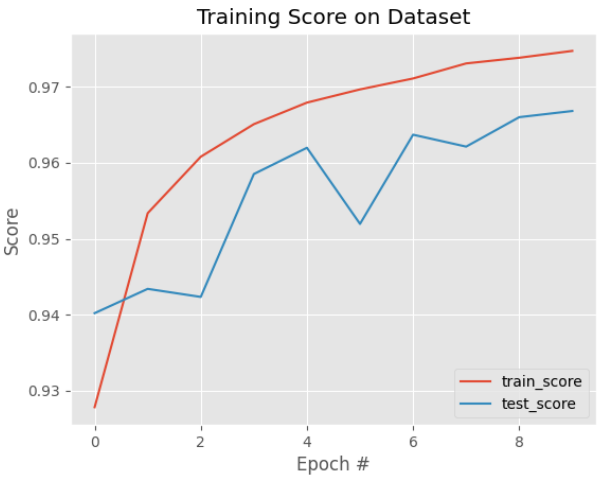
❖ ***Resnet152:***



- **Experiments amb data augmentation:**

  - ***Resnet18:***

  - ***Resnet50:***



❖ ***Resnet101:***

❖ ***Resnet152:***

Annex IV – Objectius de Desenvolupament Sostenible

Grau de relació del treball amb els Objectius de Desenvolupament Sostenible (ODS).

Objectius de Desenvolupament Sostenibles	**Alto**	**Mig**	**Baix**	**No Procedeix**
ODS 1. **Fi de la pobresa.**				**X**
ODS 2. **Fam cero.**				**X**
ODS 3. **Salut y benestar.**		**X**		
ODS 4. **Educació de qualitat.**			**X**	
ODS 5. **Igualtat de gènere.**				**X**
ODS 6. **Aigua neta i sanejament.**				**X**
ODS 7. **Energia assequible i no contaminant.**				**X**
ODS 8. **Treball decent i creixement econòmic.**		**X**		
ODS 9. **Industria, innovació i infraestructures.**	**X**			
ODS 10. **Reducció de las desigualtats**.				**X**
ODS 11. **Ciutats i comunitats sostenibles**.				**X**
ODS 12. **Producció i consumo responsables.**				**X**
ODS 13. **Acció pel clima.**				**X**
ODS 14. **Vida submarina.**				**X**
ODS 15. **Vida de ecosistemes terrestres.**				**X**
ODS 16. **Pau, justícia i institucions sòlides.**				**X**
ODS 17. **Unions per aconseguir objectius.**			**X**	

Reflexió sobre la relació del TFG con los ODS més relacionats.

- **Salut y benestar,** la creació del sistema de generació automàtica de màscares previndrà als treballadors passar gran quantitat d'hores davant d'una pantalla generant-les de forma manual, per tant guanyarà en qualitat de vida i en salut visual.

- **Treball decent i creixement econòmic,** amb el nou sistema es passa d'un procediment repetitiu, manual i costós a un automàtic. D'aquesta manera el treball passa a ser eficient, i en conseqüència es tradueix en un millor funcionament de la cadena de producció a la fàbrica i un major benefici econòmic.

- **Industria, innovació i infraestructures,** en la mateixa línia que els dos punts anteriors, comporta una gran innovació a un procés industrial. Suposa un gran canvi passar d'una tasca manual a una automàtica amb els beneficis que comporta. Aquesta millora afavoreix la eficiència i eficàcia de la cadena de producció, per tant, una millora en la infraestructura de la factoria en aquest cas.

Addicionalment als 3 ODS esmentats, hem trobat 2 dos més que es poden relacionar en el treball però en menor mesura que els anteriors. Aquests són **Educació de Qualitat (ODS 4)** i **Unions per aconseguir Objectius (ODS 17)**. Respecte al **ODS 4,** tot l'estudi i aprenentatge que ha comportat la creació d'un sistema d'aquestes característiques conformen una educació en el camp de la intel·ligència artificial de gran qualitat. Per últim, en relació al **ODS 17,** el nou sistema implica la necessitat de crear una comunicació dels operaris de la factoria amb els tècnics encarregats de mantenir el sistema sempre operatiu per tal reportar possibles errades o observacions per millorar el sistema. D'aquesta forma es crea una col·laboració de dues parts per aconseguir un objectiu comú.

La resta de ODS s'ha considerat que no procedeixen en el present treball, ja que no estan relacionats en l'àrea d'investigació del treball i la tecnologia desenvolupada no té cap efecte sobre ells.