

SENSOR-IA-L:

Sistema de Visió Avançada pel Monitoreig de Líquids Químics en la Galvanotècnia

Sergi Sanz Orellana

Resum

Aquest Treball de Final de Grau presenta el desenvolupament d'un sistema de monitoratge avançat per a línies de producció química mitjançant tècniques de visió per computador i machine learning amb col·laboració amb l'empresa Chemplate Materials, com a objectiu de supervisar cubetes amb líquids químics per detectar anomalies dins d'un procés de galvanització. Per a això, s'ha creat tot un projecte d'IoT on s'han utilitzat càmeres connectades a Raspberrys Pi Zero 2, capturant més de 100.000 imatges que s'han organitzat en un dataset per posteriorment la creació de tres models de classificació d'imatges per a tres estacions de la línia en concret. S'han entrenat diversos models de machine learning amb l'ús de transfer learning, seleccionant finalment el model Xception per la seva eficiència en la classificació d'imatges. La interfície del sistema, desenvolupada com una pàgina web, permet el control i la visualització remota de les càmeres i dels resultats de la classificació.

Paraules clau— Visió per Computador, Intel·ligència Artificial, Aprenentatge Profund, Xarxes Neuronals, Classificació de Imatges, Reconeixement de Anomalies, Planta Pilot, Galvanotècnia, Càmera, Algorismes de Predicció, Monitorització.

Abstract

This Final Degree Project presents the development of an advanced monitoring system for chemical production lines using computer vision and machine learning techniques, in collaboration with Chemplate Materials. The aim is to supervise tanks containing chemical liquids to detect anomalies within a galvanization process. An entire IoT project was created, utilizing cameras connected to Raspberry Pi Zero 2 units, capturing over 100,000 images organized into a dataset for the subsequent creation of three image classification models for specific stations in the line. Several machine learning models were trained using transfer learning, ultimately selecting the Xception model for its efficiency in image classification. The system interface, developed as a web page, allows remote control and visualization of the cameras and classification results.

Index Terms— Computer Vision, Artificial Intelligence, Deep Learning, Neural Networks, Image Classification, Anomaly Detection, Pilot Plant, Camera, Prediction Algorithms.

1 INTRODUCCIÓ - CONTEXT DEL TREBALL

En el context en el que es troba la creació d'aquest treball de fi de grau d'enginyeria informàtica de la Universitat Autònoma de Barcelona, és el de les indústries químiques en col·laboració amb l'empresa Chemplate Materials. Aquesta empresa és especialitzada en la distribució i manufactura de productes químics semielaborats per a aplicacions en sectors com la galvanotècnia. La supervisió i el manteniment de la qualitat dels processos de producció són un dels factors crítics per a garantir l'eficàcia, la seguretat i la fiabilitat de les operacions.

Aquest projecte es troba en el marc d'un prototip d'una línia de cubetes on hi ha certes cubetes que tenen líquids químics que es volen monitoritzar amb tècniques de visió per computador i aprenentatge automàtic.



Fig. 1. Planta pilot de Chemplate ubicada en nau externa.
Font pròpia.

-
- E-mail de contacte: 1599507@uab.cat
 - Menció realitzada: Computació
 - Treball tutoritzat per: J. Vázquez Corral (Ciències Computació)
 - Curs 2023/24

Les plantes pilot de Chemplate, que serveixen com a entorns crucials per al desenvolupament i la prova de tractaments químics innovadors, requereixen una supervisió contínua i precisa per a garantir el funcionament òptim dels processos. En un d'aquests processos de recobriment de peces en la galvanotècnia, es vol desenvolupar un sistema avançat de monitoratge i detecció d'anomalies amb l'ús de varies càmeres.

Amb el context clar, crearem el disseny i la implementació d'un sistema de visió avançat que utilitzi una combinació de càmeres de alta definició, plataformes computacionals i algorismes de classificació.

Ha de ser capaç de detectar i classificar canvis en aquests líquids, canvis com: la presència d'escuma, el bombogeig, la introducció de peces, entre d'altres. Amb l'objectiu de monitoritzar la línia i garantir la seguretat del procés, ja que aquests poden indicar problemes en el procés de producció que afecten la seguretat i eficàcia de la planta.

Aquesta tasca es tot un repte, que requereix una combinació de tecnologies d'adquisició d'imatges, processament d'imatges, algorismes per a la classificació, juntament amb el disseny d'una interfície d'usuari intuïtiva per a la supervisió i control del sistema.

1.1 Estructura del document

Aquest projecte, en format d'article, segueix la següent estructura: comença per la introducció, on es proporciona el context del treball, els objectius i la planificació. Continua amb, "l'estat del art", s'adreça als fonaments teòrics, amb apartats que aborden algorismes de classificació d'imatges, xarxes neuronals convolucionals, l'èxit d'aquestes i els vision transformers.

La metodologia és presentada en la tercera secció, seguida del desenvolupament, els resultats obtinguts, i finalment, les conclusions. Els agraïments, bibliografia i un apèndix completen aquesta estructura, proporcionant una visió global i detallada del projecte.

1.2 Objectius

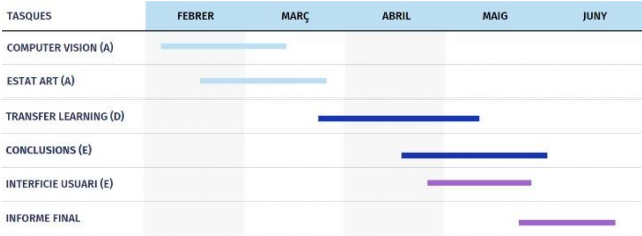
Per a du a terme amb èxit aquesta implementació és important definir uns objectius clars i factibles, els podem subdividir en 5 objectius específics:

- A. Aprendre, familiaritzar-me sobre el camp de la visió per computador, també sobre deep learning i fer un estudi del art de la classificació multi classe d'imatges.
- B. Desenvolupar un sistema de captura d'imatges utilitzant tres càmeres connectades a varies plataformes computacionals per monitorar les cubetes de la línia.

- C. Creació de un ampli dataset amb tots el casos possibles d'estat i de anomalies.
- D. Creació i implementació d'algorismes de classificació d'imatges per a la detecció d'anomalies, per mitjà de tècniques d'aprenentatge profund.
- E. Desenvolupar una interfície d'usuari intuïtiva per visualitzar l'estat del sistema, les imatges capturades i els resultats de la classificació. Per exemple pàgina web.

1.3 Planificació

La planificació es crucial ja que ens permet crear un camí a traçar per assolir els objectius.



2 ESTAT DEL ART

El camp de la classificació d'imatges, es actualment un dels camps més rellevants de la visió per computador. Però abans d'aprofundir en el concepte de classificació d'imatges, caldria reflexionar sobre la següent qüestió: Per què és crucial la classificació d'imatges?

Ens trobem immersos en l'era de la informació. Amb l'expansió de l'Internet de les Coses (IoT) i l'augment de l'ús de la Intel·ligència Artificial (IA), el món genera una immensa quantitat de dades. Aquestes dades poden prendre formes diferents però les imatges, representen una part fonamental en la generació d'aquesta informació a escala global. En essència, és la tasca d'assignar una etiqueta a una imatge a partir de un conjunt predefinit de categories.

Els models de deep learning han demostrat ser més precisos i efectius que els mètodes tradicionals de classificació d'imatges. Gràcies a la capacitat de l'aprenentatge automàtic per aprendre característiques d'un conjunt de dades, els models poden identificar patrons i característiques per classificar les imatges.

La classificació d'imatges amb l'ajuda de la IA és un àmbit molt actiu de recerca i desenvolupament, i s'està utilitzant en diverses aplicacions. Aquest ampli camp abasta una gran varietat de tipus d'imatges, des de simples fotografies fins a imatges mèdiques complexes.

Pot haver-hi confusió al voltant dels termes "xarxes

neuronal" i "deep learning", ja que sovint s'utilitzen indistintament. L'aprenentatge profund (deep learning) fa referència específicament a l'ús de xarxes neuronals amb múltiples capes, d'aquí el terme "profunditat" per denotar la profunditat de l'arquitectura de la xarxa.

El Deep learning es una tècnica de Machine learning que utilitza xarxes neuronals profundes amb moltes capes per fer prediccions. D'aquesta manera s'aconsegueix un aprenentatge de extrem a extrem, és a dir, les imatges són les entrades i tant les característiques com la classificació s'aprenen directament a partir de les imatges d'entrada.

Tot aprenentatge profund implica xarxes neuronals, però no totes les xarxes neuronals són models d'aprenentatge profund. Una xarxa neuronal amb només unes poques capes normalment no es consideraria un model d'aprenentatge profund, mentre que una xarxa neuronal amb múltiples capes ocultes s'inclouria en la categoria d'aprenentatge profund [1].

2.1 Algorismes de Classificació d'Imatges:

Els algorismes utilitzats en la classificació d'imatges són fonamentals per a l'èxit del procés. Els 3 més comuns són:

2.1.1 Màquines de Vectors de Suport (SVM):

Els SVM són àmpliament utilitzats en la classificació d'imatges degut a la seva capacitat per trobar el hiperplà que millor separa les classes en l'espai de característiques. Utilitzen vectors de suport per definir aquest hiperplà, que maximitza el marge entre les classes. Els SVM poden ser eficaços especialment en conjunts de dades amb característiques linealment separables.

2.1.2 Arbres de Decisió:

Aquests algorismes utilitzen una estructura d'arbre per classificar objectes en diferents categories. Cada node de l'arbre representa una característica de la imatge, i les branques representen les possibles sortides basades en aquesta característica. Els arbres de decisió són fàcils d'interpretar i poden manejar tant dades numèriques com categòriques.

2.1.3 Xarxes Neuronals Convolucionals (CNN):

Les xarxes neuronals són un model computacional inspirat en l'estructura i funció del cervell humà respecte a les neurones. Consisteixen en nodes interconnectats organitzats en capes. Aquestes xarxes es poden utilitzar per a diverses tasques, incloent classificació, regressió, reconeixement de patrons i més.

Són particularment efectives per al processament d'imatges degut a la seva capacitat per extreure característiques rellevants d'elles. Utilitzen capes convolucionals que apliquen filtres a la imatge per detectar patrons com a marges, textures i formes. Després, aquestes característiques passen a través de capes de pooling per reduir la dimensionalitat i finalment es connecten a capes denses per a la classificació.

Sorgeix una qüestió interessant entre aquests dos últims algorismes: una xarxa neuronal realment és un arbre de decisió molt extens?

Doncs bé, aquesta problemàtica es molt ben estudiada en el paper "Neural Networks are Decision Trees" de Çağlar Aytekin [2], on s'estudia en profunditat els dos algorismes i s'acaba concluint que realment les xarxes neuronals es poden representar equivalentment com a arbres de decisió, on aquesta representació és d'equivalència i no una aproximació.

2.2 Xarxes Neuronals Convolucionals en detall:

2.2.1 Xarxes neuronals convolucionals:

Les xarxes neuronals convolucionals, també conegudes com a CNN (Convolutional Neural Networks), són especialment adequades per a tasques de processament d'imatges. Utilitzen capes de convolució per extreure característiques visuals importants, com ara vores, formes i textures, la qual cosa les fa ideals per a la classificació multi classe d'imatges, detecció d'objectes i seguiment de moviment.

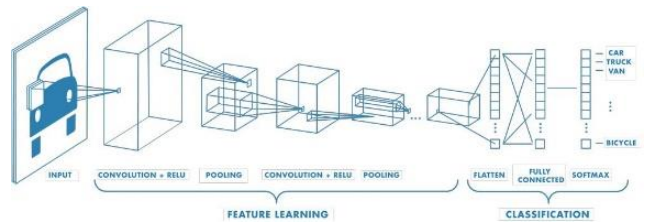


Fig. 2. Esquema visual de una CNN. Font Matlab [8].

2.3 Èxits de Xarxes neuronals convolucionals:

En l'última dècada, el desenvolupament d'arquitectures de xarxes neuronals convolucionals ha estat fonamental per a l'avanç en tasques de visió per computador, particularment en la classificació d'imatges. Algunes de les arquitectures més influents en aquest camp han sigut: VGG16, ResNet i Inception. Creades per les universitats més importants o empreses tecnològiques més grans del món.

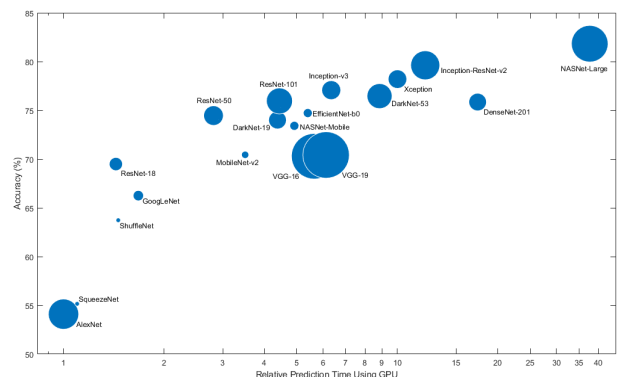


Fig. 3. CNNs per a la classificació de imatges. Font Matlab [8].

2.3.1 Model VGG:

VGG, que representa al Visual Geometry model, és un model proposat per investigadors de la Universitat d'Oxford el 2014 per a tasques de classificació d'imatges mitjançant xarxes neuronals convolucionals [X]. VGG va destacar la importància de la profunditat en les CNNs, demostrant que una xarxa profunda amb filtres 3x3 pot aprendre característiques complexes. Abans de VGG, les CNNs solien tenir menys capes i filtres més grans, mentre que VGG va mostrar que una arquitectura senzilla amb filtres 3x3 podia rivalitzar amb models més complexos [3].

És una arquitectura influent en el desenvolupament de CNNs per a visió per computador, destacant la importància de la profunditat, els filtres petits i l'estructura en capes per a l'aprenentatge de característiques complexes en imatges. Tot i que VGG va aconseguir resultats destacats en classificació d'imatges, la seva profunditat i cost computacional són desafiaments importants.

2.3.2 Model ResNet:

ResNet, abreujament de Xarxa Residual (Residual Network), va ser proposat per investigadors de Microsoft el 2015 per abordar el problema de la desaparició del gradient en xarxes neuronals molt profundes [4]. Aquest problema sorgeix en el entrenament de xarxes profundes, on els gradients disminueixen exponencialment mentre es propaguen cap enrere a través de moltes capes durant l'entrenament, dificultant que les capes primeres aprenguin de manera efectiva.

Les arquitectures de ResNet consisteixen en apilar múltiples blocs residuals, permetent la construcció de xarxes molt profundes amb més de 100 capes. ResNet va introduir el concepte d'aprenentatge residual utilitzant connexions de salt, conduint a un entrenament més ràpid i fàcil de xarxes molt profundes i aconseguint resultats d'última generació en tasques com la classificació d'imatges.

2.3.3 Model Inception:

Desenvolupada per Google Research al 2015 [5], Inception V3 és una evolució de les arquitectures Inception anteriors. Utilitza mòduls d'Inception, que consisteixen en conjunts de convolucions de diferents mides i pooling, concatenant les sortides. Es combinen múltiples camins paral·lels de convolucions i pooling per extreure característiques a diferents escales. Aconsegueix un equilibri entre rendiment i eficiència computacional, sent capaç de manejar imatges d'alta resolució de manera efectiva.

En resum, les arquitectures VGG16, ResNet i Inception V3 han estat crucials en l'avanç de la visió per computadora. Cadascuna d'aquestes arquitectures té les seves pròpies característiques distintives i desafiaments, i la seva elecció depèn del problema específic que s'estigui abordant i les restriccions computacionals disponibles.

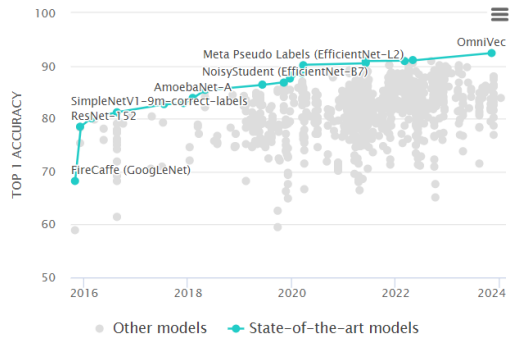


Fig. 4. Estat del art dels models per a la classificació de imatges basat del dataset famós Imagenet. Font PapersWithCode [6].

En resum, les xarxes neuronals convolucionals són una eina fonamental en l'aprenentatge automàtic. Comprendre els diferents tipus de xarxes neuronals i les seves aplicacions és essencial per aprofitar al màxim el seu potencial en una àmplia varietat de problemes.

De les xarxes neuronals basades en les CNNs van evolucionar a una nova tecnologia presentada en el 2021 que detallaré a continuació.

2.4 Vision Transformer (ViT):

Introduït per Dosovitskiy et al. el 2021 [7], els Vision Transformers (ViTs), són una nova arquitectura de xarxa neuronal que està revolucionant el camp de la visió per computador. A diferència dels models convolucionals, que han dominat la visió per computador durant anys, els ViTs fan servir transformadors, una arquitectura inicialment dissenyada per a tasques de processament de llenguatge natural, aquesta son la base dels models de intel·ligència artificial.

Els Vision Transformers funcionen segmentant la imatge d'entrada en petits fragments, que són aplanats i introduïts en una arquitectura de transformador. En lloc d'utilitzar capes convolucionals, els transformadors processen aquests fragments utilitzant matrius d'atenció per aprendre les relacions entre els píxels. Aquest enfocament permet als ViTs capturar relacions de llarga distància en les imatges sense dependre de les operacions locals de les convolucions.

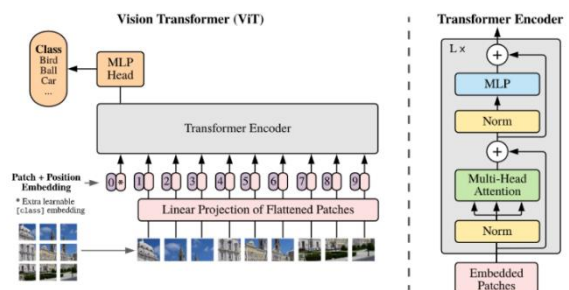


Fig. 5. ViT per a la classificació de imatges. Font paper: An image is worth 16x16 words [7].

Tradicionalment, les CNNs han estat l'arquitectura preferida per a diverses tasques de visió per computador, incloent-hi la classificació d'imatges, la detecció d'objectes i la segmentació. No obstant això, l'arquitectura Transformer, va demostrar un rendiment alt a través de l'ús de mecanismes d'auto atenció.

Però tot i així, en un dels articles més importants sobre aquest tema anomenat “An Image is Worth 16x16 Words: Transformers for Image Recognition, 2021” es diu:

“In computer vision, however, convolutional architectures remain dominant...”

Això es dit ja que s'ha pogut comprovar que els ViT son millors que les CNNs a partir de les 100 milions de imatges de entrenament i pel moment no es una millora molt significativa, per al cost computacional que impliquen. En resum, els Vision Transformers estan canviant la manera en què abordem la classificació d'imatges, proporcionant una alternativa innovadora i prometedora a les CNNs.

Amb tot aquest estudi del art clar, considero que la metodologia més adient per portar a terme aquest projecte és la creació de varies xarxes neuronal convolucionals a partir de xarxes neuronals preentrenades aplicant l'ús de la metodologia Transfer Learning [10]. Una vegada fet això, escollir el millor resultat per a la aplicació del projecte.

3 METODOLOGIA

En aquesta part del treball es posaran en pràctica alguns dels coneixements desenvolupats anteriorment. L'objectiu és crear tres classificadors d'imatges multi classe que identifiquin amb una alta precisió en quin estat es troba la cubeta.

Per a du a terme això s'utilitzaran conceptes de deep learning per realitzar el codi en python amb la llibreria Keras on es crearan els models per fer els classificadors a partir de xarxes neuronals ja preentrenades.

3.1 Dataset:

Per poder crear un bon classificador, és molt important el conjunt de dades o dataset. Com més equilibrat sigui aquest conjunt de dades i més representatiu sigui, més precisió s'obtindrà. Una vegada obtingut es dividirà entre un 85% del data set com a entrenament, un 15% com a validació.

3.1.1 Preprocessament de les dades:

El preprocessament de dades és essencial per garantir que les dades estiguin en el format adequat i siguin de qualitat abans d'entrenar el model. Algunes tècniques comunes:

- **Redimensionament d'Imatges:** Ajustar totes les imatges a una mida comuna facilita el processament i redueix la variabilitat en el conjunt de dades, es sol utilitzar un ajustament quadrat de 224x224 o de 229x229 depenent de la arquitectura utilitzada.
- **Normalització d'Imatges:** Assegurar que els valors dels píxels estiguin en un rang específic, com 0 a 1, ajuda a estandarditzar les dades i facilita el procés d'entrenament, el mateix amb el color, si una imatge es a escala de grisos s'ha de transformar a imatges RGB amb els 3 canals de color.
- **Augment de Dades:** Generar noves imatges a partir de les existents mitjançant transformacions com rotacions, desplaçaments o canvis en el contrast augmenta la diversitat del conjunt de dades i ajuda a prevenir el overfitting.

3.1.2 Entrenament del model:

L'entrenament del model implica alimentar-lo amb dades etiquetades i ajustar els seus paràmetres perquè pugui realitzar prediccions precises sobre noves imatges. Aquest procés generalment implica iteracions d'ajust dels paràmetres del model fins que aconsegueixi un nivell satisfactori de precisió, ajustant el batch size i les èpoques.

3.1.3 Avaluació del rendiment:

L'avaluació del rendiment del model es realitza utilitzant mètriques com l'Accuracy i el Loss. Aquestes mètriques permeten comprendre com de bé està realitzant les classificacions el model i quantes classificacions correctes realitza en relació amb les classificacions incorrectes de manera general.

3.1.4 Transfer Learning:

És molt pràctic començar amb una xarxa preentrenada, com ara MobileNetV2. Però manca de flexibilitat en quant a la forma en què funciona la xarxa, i no resoldrà el problema concret que volem resoldre. Es pot crear i entrenar una xarxa des de zero, començant només amb l'arquitectura de la xarxa i pesos aleatoris.

Però per a aconseguir bons resultats requereix molt d'esforç: (1) coneixements i experiència en arquitectura de xarxes (2) una gran quantitat de dades d'entrenament i (3) molt temps de càlcul i recursos d'alt rendiment com una GPU.

La transferència d'aprenentatge és una solució eficient per a molts problemes. La formació requereix algunes dades i temps de càlcul, però és més ràpida que començar de zero, i el resultat és una xarxa adaptada específicament al nostre problema.

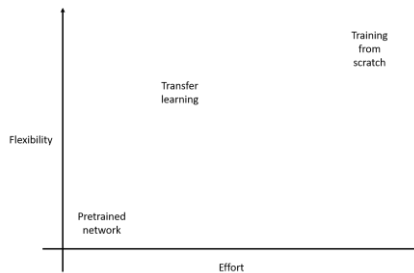


Fig. 6. Gràfica esforç versus flexibilitat. Font Matlab [8].

Per dur a terme transfer learning s'ha de reemplaçar la capa de sortida del model preentrenat i reentrenant-lo amb dades específiques del nou domini. Faré la comparació entre els resultats de les xarxes MobileNetV2, VGG-19, Resnet-50, InceptionV3 i Xception.

Model	Size (MB)	Top-1 Accuracy	Top-5 Accuracy	Parameters	Depth	Time (ms) per inference step (CPU)	Time (ms) per inference step (GPU)
Xception	88	79.0%	94.5%	22.9M	81	109.4	8.1
VGG16	528	71.3%	90.1%	138.4M	16	69.5	4.2
VGG19	549	71.3%	90.0%	143.7M	19	84.8	4.4
ResNet50	98	74.9%	92.1%	25.6M	107	58.2	4.6

Fig. 7. Els 4 millors models desde Tensorflow. Font Keras [10].

Amb l'ajuda de Tensorflow i Keras podrem fer això de manera molt fiable, ja que te carregats en el programa les xarxes neuronals més famoses i amb l'ús de Python podem modificar les últimes capes per a que s'ajusti al nostre dataset.

3.1.5 Proves exhaustives i comparació de resultats

Realitzar proves exhaustives implica avaluar el rendiment del model en diferents escenaris i condicions per garantir la seva eficàcia i robustesa. Això pot incloure la variació en la il·luminació, l'angle de visió, la mida de la imatge, entre altres factors, per assegurar-se que el model pugui generalitzar bé a situacions reals en la planta pilot de Chemplate.

4 DESENVOLUPAMENT

El primer pas és la elecció de la càmera, després de haver fet un estudi sobre les possibilitats que hi ha en el mercat i ajustant-me al pressupost de l'empresa, m'he decantat per la càmera ELP Pro USB4k, ideal per projectes de visió per computador.

Degut a que te connectivitat per USB i és una de les millors càmeres industrials del mercat, destacar la seva qualitat de imatge i de vídeo, tan de resolució com de color i el seu zoom digital de fins 5x indispensable per enquadrar la imatge al nostre gust. Com hem de monitoritzar tres cubetes, hem comprat tres.



Fig. 8. Càmera ELP Pro USB4k. Font pròpia.

Aquestes càmeres per si soles no poden fer res, necessiten d'un ordinador per ser controlades. La manera més adequada es connectar-les a unes Raspberrys Pi Zero 2 [11]. Es tracten d'un petit ordinador de placa única que està dissenyat per ser extremadament compacte i assequible, però encara ofereix un bon rendiment per a una varietat de projectes d'electrònica, informàtica i programació.

Algunes de les seves característiques inclouen un processador ARM de 4 nuclis, 512 MB de RAM, connectivitat Wi-Fi i Bluetooth, ports USB, per connectar perifèrics i expansions, i compatibilitat amb una àmplia gamma de sistemes operatius, incloent-hi diverses distribucions de Linux. També hem comprat 3 per a que cada càmera funcioni de manera independent.



Fig. 9. Raspberry Pi Zero 2W [11].

El següent pas és poder connectar-nos a les càmeres remotament per mitjà de les Raspberry pi Zero 2. Per a poder accedir remotament he utilitzat la plataforma VNC Viewer [12] que permet connectar-se a una IP concreta, però amb la restricció de què l'ordinador i la Raspberry (en aquest cas) han d'estar connectades a la mateixa xarxa Wi-Fi.

Com la línia es troba en una altra ubicació que les oficines, l'aplicació no permet poder accedir-hi remotament. Per solucionar aquest problema, el que he fet és crear una VPN per a cada Raspberry i pel meu ordinador amb l'aplicació DrayTek [13].

Gràcies a això i a un petit script.sh que s'executarà automàticament quan una Raspberry estigui connectada a el corrent, es connectarà al Wi-Fi de la nau on està la línia i directament encendra la VPN i així podrem accedir remotament des de qualsevol lloc a les càmeres.

A més a més, per tenir una altra línia d'accés també podrem entrar per ssh gràcies a MobaXterm [15].

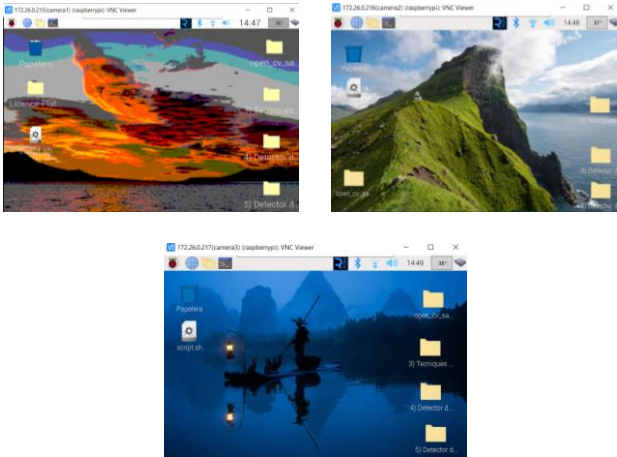


Fig. 10. Les 3 Raspberry Pi Zero 2, controlades remotament amb VNC Viewer [12]. Font pròpia.

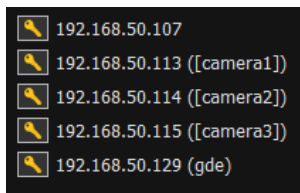


Fig. 11. Accés per SSH per MobaXterm [15]. Font pròpia.



Fig. 12. Esquema de la connectivitat del sistema. Font pròpia.

Una vegada tenim resolta la connectivitat remota de les càmeres i les Raspberry amb l'ordinador personal el següent pas es considerar l'estructura on hi estaran les càmeres fixes per a demanar la seva creació a una empresa externa capaç de fer-la:

- L'estructura ha de ser independent a la línia pilot (es recolzarà directament al terra).
- L'estructura haurà de tenir una barra regulable en alçada entre 1,70 i 2,20cm.
- L'estructura ha de permetre la fixació de càmeres en la seva estructura. Ha de ser robusta i segura.
- No es contempla cap tipus de sistema per desplaçament de les càmeres.



Fig. 13. Estructura muntada en la planta pilot de Chemplate. Font pròpia.

Amb aquesta estructura que passarà per sobre de la línia de les cubetes quedaran fixades les tres càmeres, ens interessa monitoritzar amb visió per computador la cubeta 2, 3 i 5. La primera apuntant a la cubeta 2, on hi ha un bany d'aigua per netejar la peça després de passar per la cubeta 1, i la segona apuntant a la cubeta 3, on hi ha líquid nítric de decapat per a peces d'alumini i la tercera en la cubeta 5, on hi ha aigua desionitzada.

El flux d'una peça que passa per la línia és:

- Cubeta 1 de desengreix.
- Cubeta 2 bany d'aigua.
- Cubeta 3 decapat per a peces d'alumini.
- Cubeta 4 dutxa d'aigua.
- Cubeta 5 de aigua desionitzada.



Fig. 14. Càmera i Raspberry col·locades a l'estructura construïda. Font pròpia.

Una vegada aquests passos realitzats, càmeres col·locades i connectades a les Raspberry, es el moment de fer experiments i crear el nostre dataset. Per mitjà d'un fitxer creat amb Python executarem el programa per fer la captura de imatges i aquestes imatges es guardaran en carpetes segons el dia i la hora exacta de la realització de la fotografia.

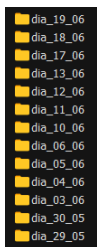
Aquest programa està creat fent una API per a que pugui ser cridat i al començar l'experiment comenci la captura d'imatges i al finalitzar l'experiment es pari aquesta, així la persona dedicada a fer els experiments, només li ha de donar al botó de començament i finalització.

És important tenir una bona quantitat de fotografies, però sobretot moltes imatges diferents, on es vegin clars els

events que poden ocórrer. Per lo tant hem de provocar aquests canvis que poden succeir per tenir una bon ventall de imatges i així poder crear un bon model entrenant-lo amb aquestes imatges. Els experiments per la captura d'imatges es farà durant el mes de maig i juny es la següent.

Les imatges es guardaran dins de cada Raspberry que tenen una capacitat de 128GB, i cada dia a les 21h les carpetes amb les imatges seran enviades automàticament per mitjà de ssh a un servidor per pre-processarles i manualment crear el dataset.

El dataset ha sigut creat a partir de més de 10 experiments i més de 100.000 imatges. Una vegada ordenat i reduït ha quedat de la següent manera: tenim 3 carpetes una carpeta per cada càmera i dins de cada carpeta, ordenat per els estats que pot tenir cada cubeta:



<ul style="list-style-type: none"> dataset_camera1 dataset_camera2 dataset_camera3 <p><i>Dataset de les 3 càmeres</i></p>	<ul style="list-style-type: none"> ALT_NIVELL_ESCUMA_CUBETA2 AMB_TAPA_CUBETA2 BAIX_NIVELL_ESCUMA_CUBETA2 BOMBOGEIG_CUBETA2 PECES_SUBMERGIDES_CUBETA2 PECES_SUBMERGIDES+BOMBOGEIG_CUBETA2 PECES_SUBMERGIDES+ESCUMA_CUBETA2 SENSE_TAPA_CUBETA2 <p><i>Dataset càmera 1</i></p>
<ul style="list-style-type: none"> AGITACIO_ACTIVADA_CUBETA3 AMB_TAPA_CUBETA3 MIG_TAPA_CUBETA3 PECES_SUBMERGIDES_CUBETA3 PECES_SUBMERGIDES+AGITACIO_CUBETA3 PECES_SUBMERGIDES+MIG_TAPA_CUBETA3 SENSE_TAPA_CUBETA3 <p><i>Dataset càmera 2</i></p>	<ul style="list-style-type: none"> AGITACIO_ACTIVADA_CUBETA5 AMB_TAPA_CUBETA5 PECES_SUBMERGIDES_CUBETA5 PECES_SUBMERGIDES+AGITACIO_CUBETA5 SENSE_TAPA_CUBETA5 <p><i>Dataset càmera 3</i></p>

La Raspberry Pi Zero 2 té la capacitat d'executar una varietat de programes, inclosos diversos sistemes operatius. El sistema operatiu més utilitzat és el sistema operatiu oficial, que és el Raspberry Pi OS basat en Linux. El programes i les APIs relacionades amb l'aprenentatge automàtic i profund, com ara Python, TensorFlow, OpenCV, etc., es poden instal·lar a la Raspberry Pi i es poden utilitzar com a dispositiu per a executar inferències.

La construcció i entrenament dels models és particularment difícil d'implementar en una Pi a causa dels recursos limitats. Per això els models els crearé i s'entrenaran en un sistema molt més potent, com ara un PC, una vegada creats per executar-los a la Raspberry Pi requereix pre-processament i conversió a TensorFlow Lite.

Per crear els models utilitzarem Python i les llibreries Keras, Tensorflow i Tensorflow Lite. Es crearan des de el ordinador amb un processador intel core i5, i com hem

comentat anteriorment farem una comparativa dels models creats amb transfer learning dels 5 models més famosos i rellevants.

Que son els següents: VGG19, Resnet50, InceptionV3, MobileNetV2 i Xception.

Necessitarem tres models, un per cada Raspberry ja que com apunten a cubetes diferents, no es pot reutilitzar el mateix model, escolliré el model que millors resultats obtingui.

El següent pas es crear una interfície estil pàgina web per al control i visualització de les càmeres. Aquesta pagina web s'executarà a la raspberry servidor web i es podrà accedir des de qualsevol navegador afegint la IP de la raspberry.

Per a poder accedir a aquesta web, el que s'ha de fer es escriure al navegador la ip del servidor on esta la web executant-se aquesta web recull les web de cada Raspberry i així queda centralitzat, per poder entrar solament amb la ip i per a que estigui disponible a totes hores el que s'ha fet es pujar-ho a un serveix de producció amb Unicorn i Nginx.

Des d'aquesta interfícies es podrà inicialitzar la captura d'imatges, frenar la mateixa, executar el model creat, obrir la càmera, tancar-la, entre d'altres. Està feta amb Python amb l'ús de la llibreria Flask i amb Html i Css.



Fig. 15. Interfície web pàgina principal. Font pròpia.

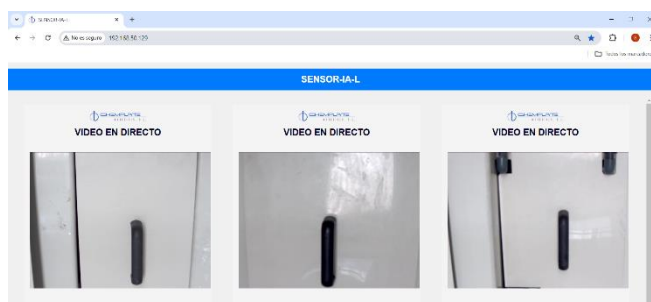
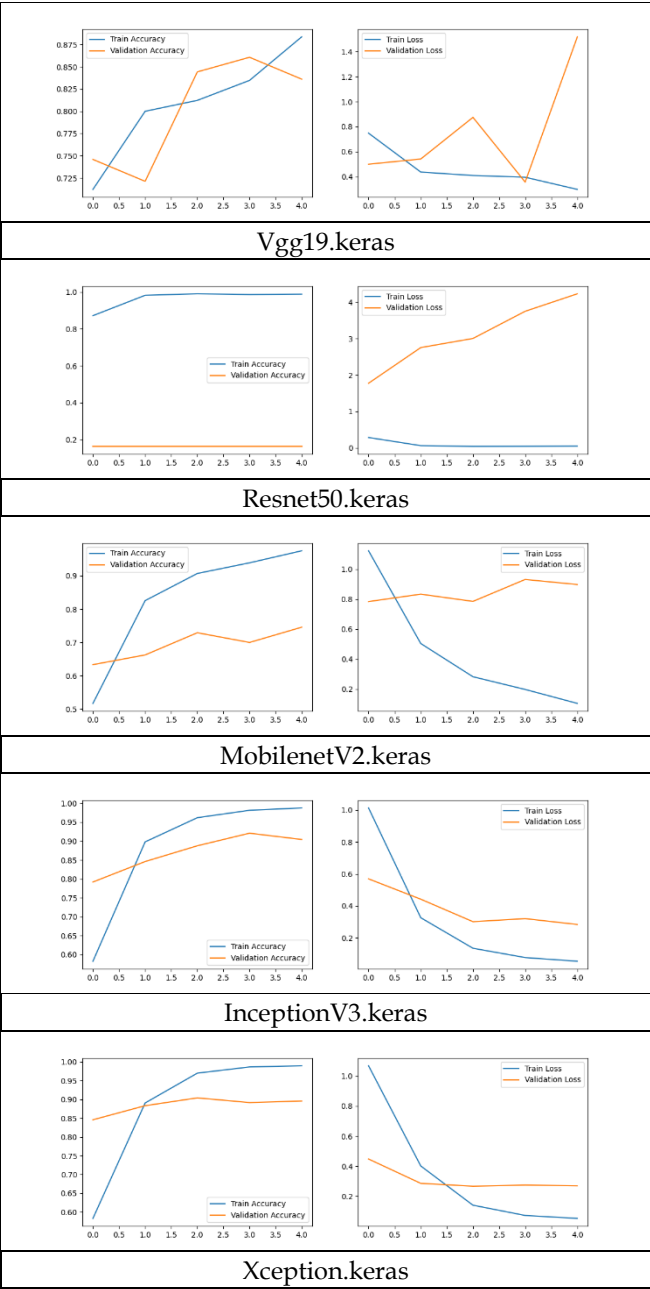


Fig. 16. Interfície web visualització càmeres. Font pròpia.

5 RESULTATS

Aquests son els resultats del entrenament fent transfer learning amb el dataset obtingut en aquestes setmanes d'experiments. Després de varies modificacions aquestes son les millors gràfiques aconseguides respecte l'acuraccy i el loss amb un entrenament de 5 èpoques en el model per la càmera 1:



Podem observar que amb una base com Vgg19 obtenim bons resultats tant d'entrenament com de validació i no en tenim overfitting. Event que si ens succeeix amb Resnet50, on tenim molt bons resultats en el entrenament i pèssims en la validació.

Amb la xarxa neuronal creada a partir de MobileNet patim també d'overfitting i un Loss alt i on trobem els millors resultats son en les xarxes neuronals InceptionV3 i Xception. On l'acuraccy es molt bó i no patim d'overfitting.

Es per això que ens quedarem amb la xarxa neuronal creada a partir de Xception per a les tres càmeres i convertirem els models creats de .keras a .tflite per poder executar-los en les raspberrys. Fent una interfície web amb Flask, Python, Html, Css i Javascript, acabem tenint això:

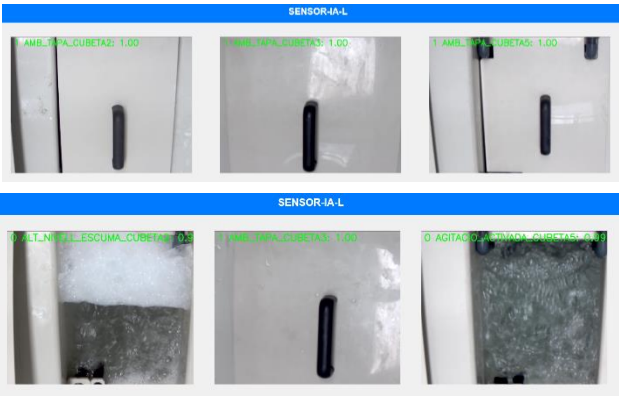


Fig. 17. Captures de la classificació en temps real. Font Pròpia.

Tant per els tres models la predicció es molt fiable i robusta, i no decau el rendiment ni la memòria de les Pi. Així que repte aconseguit, ara el que continua es fer un històric en temps real de la predicció que fa en cada moment. Per a això enviarem a una base de dades de PostgreSQL[16] la predicció feta en cada moment.

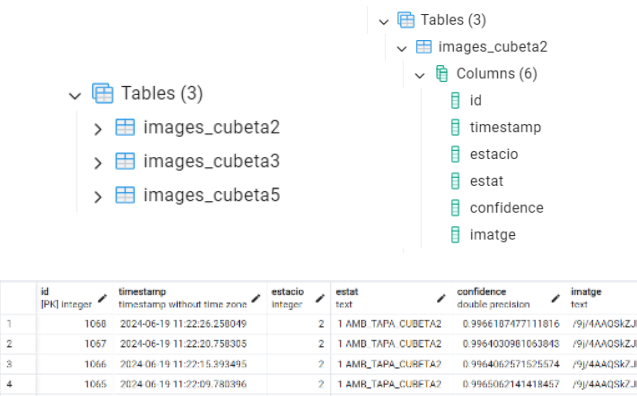


Fig. 18. Captures de la Base de Dades en temps real. Font Pròpia.

D'aquesta manera, es guardarà cada 5 segons la predicció feta de cada càmera en la base de dades per poder tractar-les posteriorment.

6 CONCLUSIONS I MILLORES

Un problema de classificació d'imatges en xarxes neuronals és una tasca en la qual s'entrena un model d'aprenentatge automàtic per assignar etiquetes o categories a imatges segons el seu contingut visual. L'objectiu és que el model aprengui a reconèixer i distingir automàticament els diferents objectes, patrons o característiques presents a les imatges i els associï amb les etiquetes correctes.

Així mateix, mitjançant l'aprenentatge per transferència, que és una tècnica senzilla però poderosa, es pot aprofitar el coneixement previ desenvolupat per xarxes neuronals com les visualitzades en aquest treball, on les representacions apreses en relació amb els problemes de classificació permeten millorar l'enteniment de problemes de magnitud similar d'una manera adequada.

Per tenir bons resultats en els classificadors creats, és importantíssim un bon data set amb abundància d'imatges i ben ordenades per cada categoria diferent.

El model Xception en aquest cas ha demostrat millors comportaments i prediccions contra els models creats amb VGG19, Resnet50, InceptionV3 i MobileNetV2.

Com a millores en un futur a curt termini o com a continuació del projecte, m'agradaria haver fet una implementació final de visualització de la classificació predita de manera atractiva, com una espècie de dashboard amb Grafana, per exemple, però no ha sigut possible per falta de temps.

Aquest projecte m'ha brindat una excel·lent oportunitat per millorar les meves habilitats professionals i aprofundir en el camp del Machine Learning, específicament en l'àrea de la visió per computador. Ha estat una experiència molt interessant, estimulante i enriquidora.

7 AGRAÏMENTS

En aquest apartat m'agradaria agrair a Chemplate Materials, concretament a Andreu Ruiz per la confiança depositada en mi i proporcionar-me els recursos necessaris per poder dur a terme aquest treball. En segon lloc al meu tutor Javier Vázquez pel seu feedback en cada entrega i en tercer lloc, a la meua família, parella i amics pel suport incondicional.

8 BIBLIOGRAFIA I FONTS D'INFORMACIÓ

- [1] IBM. (s.f.). Neural Networks. Recuperat de <https://www.ibm.com/topics/neural-networks> Visitat el: 19/04/24
- [2] Aytekin, C. (2022). Neural networks are decision trees. arXiv preprint arXiv:2210.05189.
- [3] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional

networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

- [4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).
- [5] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).
- [6] Papers With Code. (s.f.). Image Classification on ImageNet. Recuperat de <https://paperswithcode.com/sota/image-classification-on-imagenet> Visitat el: 19/04/24
- [7] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... & Houlsby, N. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929.
- [8] MATLAB. (2010). version 7.10.0 (R2023b). Natick, Massachusetts: The MathWorks Inc.
- [9] Python Software Foundation. (2020). Python 3.9. Recuperat de <https://www.python.org/downloads/release/python-390/> Visitat el: 19/04/24
- [10] Gulli, A., & Pal, S. (2017). Deep learning with Keras. Packt Publishing Ltd.
- [11] Raspberry Pi Foundation. (s.f.). Raspberry Pi Zero 2 W. Recuperat de <https://www.raspberrypi.com/software/> Visitat el: 19/04/24
- [12] RealVNC Ltd. (2024). VNC Viewer (Versió 6.21.118). <https://www.realvnc.com/en/connect/download/viewer/> Visitat el: 27/06/2024
- [13] DrayTek. (s.f.). DrayTek Smart VPN Client. Recuperat de <https://www.draytek.com/products/smart-vpn-client/> Visitat el: 27/06/2024
- [14] Ameen, S., Siriwardana, K., & Theodoridis, T. (2023). Optimizing Deep Learning Models For Raspberry Pi. arXiv preprint arXiv:2304.13039.
- [15] Moba [1] Mobatek. (s.f.). MobaXterm. Recuperado de <https://mobaxterm.mobatek.net/> Visitat el: 27/06/2024
- [16] PostgreSQL Global Development Group. (s.f.). PostgreSQL. Recuperado de <https://www.postgresql.org/> Visitat el: 27/06/2024

9 APÈNDIX

A1. GITHUB

En aquest repositori github he penjat tot el codi desenvolupat en la creació d'aquest treball de fi de grau:

<https://github.com/sergisanzchemplate/SENSOR-IA-L>