Universitat de Girona
Escola Politècnica Superior

Treball final de grau

Estudi: Grau en Enginyeria Informàtica

Títol: Generació de dades sintètiques d'entrenament d'una xarxa neuronal per a la segmentació de peixos en imatges adquirides en un entorn controlat

Document: Memòria del projecte

Alumne: Alexander Tempelaar Sánchez

Tutor: Rafael García / Ricard Prados
Departament: Arquitectura i Tecnologia de Computadors
Àrea: Arquitectura i Tecnologia de Computadors

Convocatòria (mes/any) Febrer 2020

Índex

1. Introducció, motivacions, propòsit i objectius del projecte

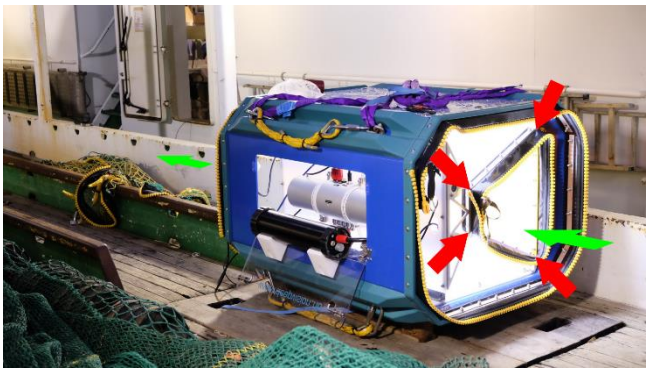
1.1 Introducció

A dia d'avui, en l'àmbit de la pesca industrial, existeix una gran quantitat de volum descartat, degut a que bona part dels espècimens capturats no són del l'espècie o mida desitjada. La sobrepesca té un impacte devastador en el medi ambient, i des de fa uns anys s'estan prenent mesures per disminuir-lo. Un fet que ho constata és la aplicació de la llei europea "*EU landing obligation"* entrada completament en vigor l'any 2019, i que prohibeix als vaixells de pesca tornar al mar la captura no desitjada abans de l'arribada al port. Allà la captura serà pesada i posteriorment processada, moment en el qual es podrà dur a terme la tria de la fracció a conservar i a descartar. L'objectiu de la normativa és la de promoure l'optimització dels processos de pesca per a fer-los més eficients i reduir-ne l'impacte ambiental.

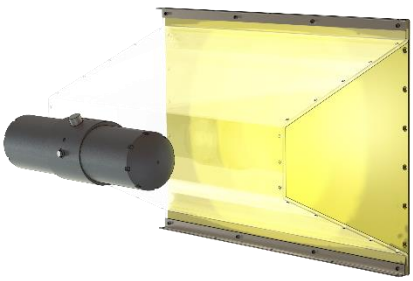
Durant els últims anys s'han utilitzat sondes nàutiques per detectar la quantitat i el tipus d'espècies presents en bancs de peixos, però la fiabilitat d'aquest sistema es troba lluny de ser perfecte. Per aquest motiu, una solució que s'està investigant actualment és la utilització de càmeres òptiques per a la monitorització detallada del tipus de peix que entra a les xarxes.

El sistema Deep Vision (Scantrol Deep Vision AS, Bergen, Noruega), és un dispositiu pioner que fa ús de càmeres òptiques i està dissenyat per ajudar a monitoritzar la pesca en els àmbits comercial i de la recerca. Aquest sistema consta d'una cambra que inclou un cilindre amb un sistema estèreo de càmeres que es col·loca a l'inici del cóp d'una xarxa, i la captura es fa passar a través de la mateixa, tal i com es pot observar a la *Figura 1*. El dispositiu Deep Vision té com a finalitat ser una eina que permeti obtenir estadístiques en temps real sobre allò que està entrant a la xarxa, i eventualment donar la possibilitat d'alliberar els peixos capturats, en cas de tractar-se d'espècimens que no tenen la mida adequada o no pertanyen a l'espècie desitjada.

(a)



(b)



(c)

Figura 1. Deep Vision Subsea System. (a) Sistema acoblat al cóp d'una xarxa de pesca. (b) Representació de la secció per on passen els peixos capturats. (c) Esquema que mostra en detall la zona de pas dels peixos, que disposa d'un metacrilat que els manté a 20 cm de les càmeres i allunyats dels focus LED.

1.2 Motivacions

Per monitoritzar la captura que entra a la xarxa de forma automàtica cal segmentar, identificar (espècimen i espècie), mesurar i fer un seguiment de cadascun dels peixos que hi passa. Actualment existeix un model de xarxa neuronal implementat sobre una arquitectura anomenada Mask R-CNN, que s'ha entrenat per aplicar segmentació d'instàncies a nivell de píxel sobre imatges obtingudes pel sistema Deep Vision, i que permet etiquetar correctament peixos que es troben aïllats dels altres presents al seu voltant, com es pot veure a la *Figura 2 (a, b)*. Una bona segmentació dels peixos permet mesurar-los, fent ús de la informació proveïda pel sistema estèreo, aplicar mètodes per fer un *tracking* dels objectes que hi apareixen. La segmentació és també el primer pas per a la classificació dels peixos en funció de la seva espècie. El problema sorgeix quan en les imatges es troben instàncies de peixos superposats, com es pot veure a la *Figura 2 (c, d)*.

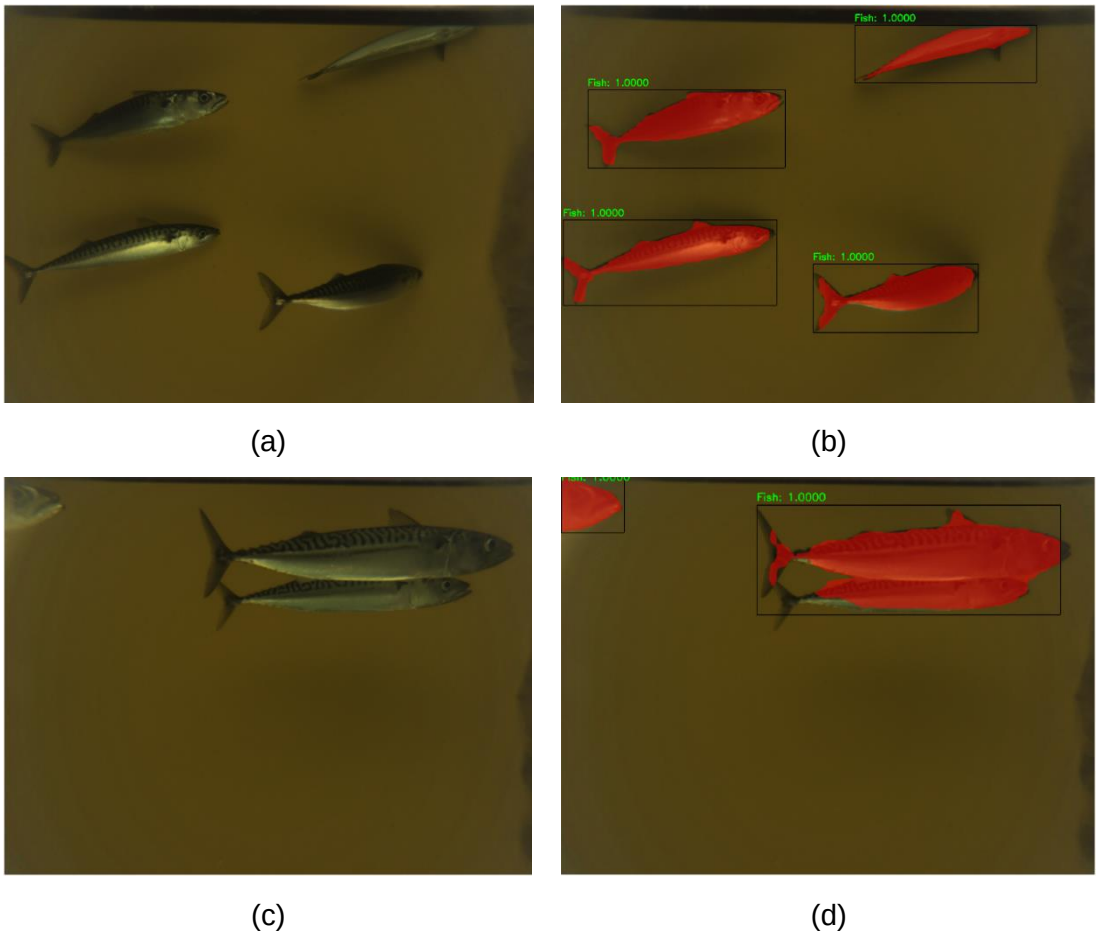


Figura 2. Resultats obtinguts amb la inferència del Mask R-CNN. (a)(c) Imatges que provenen del dataset *RVendla*. (b) Resultat que representa peixos aïllats. (d) Resultat que representa peixos amb solapament .

Gran part de la culpa de la segmentació incorrecta ve donada per una de les característiques de l'aprenentatge profund (en anglès, *Deep Learning*), i és que per poder entrenar correctament una xarxa neuronal es necessita d'una gran quantitat de imatges etiquetades que permetin ensenyar a la xarxa a segmentar correctament. La precisió i la quantitat d'imatges etiquetades representatives dels diferents escenaris possibles afecta de manera directa al rendiment de la xarxa entrenada amb aquestes.

El model del qual parteix aquest projecte es va entrenar sobre un conjunt de dades amb un nombre d'imatges reduïda, tal i com es mostra a la *Figura 3*. Si a més del nombre reduït d'imatges s'afegeix el fet que només una petita part d'aquestes inclouen peixos amb superposició. Es fa evident, doncs, que la raó per la qual la segmentació és incorrecta és l'escassetat de dades amb informació rellevant utilitzades per a re-entrenar el model.

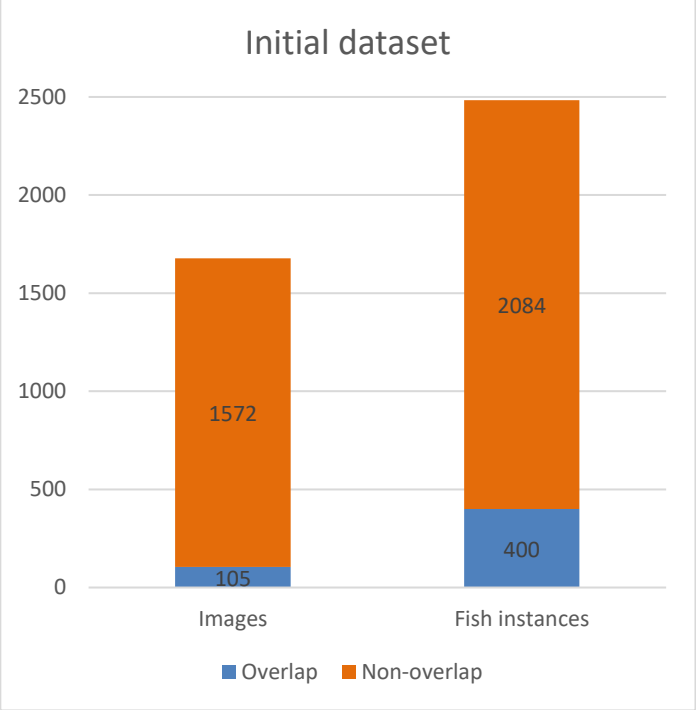


Figura 3. Representació gràfica del número d'imatges etiquetades disponibles. La gràfica mostra per una part el número d'imatges (*Images*) que presenten solapaments (*Overlap*) i els que no (*Non-overlap*). Per altre banda, en la columna de *Fish instances* es mostra el número d'instàncies de peixos que presenten solapament o no.

Al tractar-se d'imatges etiquetades a nivell de píxel, en les que es necessita una precisió important, la generació d'aquestes és un procediment lent i costós, que habitualment requereix d'un gran esforç humà i de l'ús de programes específics que permetin aquest tipus de processament, fets que dificulten en gran part l'augment del nombre d'imatges del conjunt de dades.

Per poder combatre aquesta deficiència en les dimensions dels datasets utilitzats per l'entrenament de xarxes neuronals o classificadors, s'ha dut a terme recerca en les que s'ha buscat generar imatges sintètiques que continguin prou informació rellevant i semblant a les imatges reals perquè els algorismes d'aprenentatge en puguin treure característiques (Allken *et al.*, 2019). Les millores en els resultats d'aquests treballs han conduït al plantejament d'utilitzar aquesta tècnica per solucionar el problema existent respecte la segmentació sobre imatges del sistema Deep Vision degut al les poques dades d'entrenament. El fet de que el sistema tingui una il·luminació uniforme i un color del fons conegut representa també un punt a favor en quant a processament i la generació de les noves imatges, suposant un factor important a la hora d'escollir la utilització d'aquest tipus de tècnica.

1.3 Propòsit

Aquest projecte busca abordar el problema del nombre reduït d'imatges que contenen instàncies de peixos amb superposició amb la generació d'imatges sintètiques. Aquestes noves imatges s'han de generar a partir de les dades obtingudes anteriorment de forma manual i s'espera que l'increment del numero d'aquestes dades permeti re-entrenar un model d'aprenentatge profund que sigui prou robust i capaç de distingir entre dos peixos que es solapen en imatges obtingudes pel sistema Deep Vision.

1.4 Objectius

Per poder assolir el propòsit de millorar el rendiment a l'hora de fer la segmentació d'instàncies amb la generació d'imatges sintètiques, s'ha desglossat el procediment en sis punts principals:

1.4.1 Obtenció i selecció de dades

La part més important per dur a terme el projecte és la de tenir accés a dades obtingudes pel sistema Deep Vision. En aquest cas, es disposa d'imatges que s'han obtingut d'una expedició anomenada *REDUS Vendla* (el nom del buc amb la que es va realitzat) que es va dur a terme l'any 2017. Aquestes dades s'han capturat amb la mateixa configuració de càmeres i poden tant no contenir objectes en primer com incloure quantitats elevades de peixos, gambes, bombolles i escames. L'aparició d'objectes en les imatges sol seguir un patró, com ara en el cas de les imatges sense contingut, que són obtingudes al principi i final de la pesca.

Tenint en compte la gran quantitat d'imatges a disposició, una primer part del projecte consisteix en seleccionar aquelles que contenen informació rellevant. Un exemple podria ser descartar la major part de les imatges buides i guardar-ne només unes quantes per utilitzar-les com a referència del fons.

1.4.2 Preprocessat

Aquesta part consisteix en l'estudi i l'aplicació de tècniques de processament per poder preparar les imatges per facilitar la seva manipulació posterior, ja sigui per facilitar la feina a l'algorisme de segmentació o per la generació del *groundtruth*.

1.4.3 Etiquetatge

Per tal de generar imatges sintètiques i per l'entrenament i testeig de la xarxa neuronal es necessiten imatges etiquetades. En aquest apartat es busca un procediment que ha de permetre etiquetar imatges sense dependre només d'un procés manual.

1.4.4 Generació d'imatges sintètiques

Es tracta de l'estudi i implementació del procés de generació de les imatges sintètiques a partir de imatges etiquetades prèviament. Ha d'incloure l'extracció d'instàncies de peixos de les imatges etiquetades i la modificació i inserció d'aquestes instàncies en les imatges noves.

1.4.5 Re-entrenament de la xarxa neuronal

Aquest apartat consisteix en la posta a punt de l'entorn per executar l'algorisme de aprenentatge profund en un sistema operatiu Windows i el re-entrenament de models de la xarxa neuronal Mask R-CNN utilitzant imatges reals i sintètiques. Les dades utilitzades s'han de separar en dades d'entrenament, de validació i de test. El conjunt de dades de validació i de test han d'incloure només imatges reals, mentre que les dades d'entrenament també inclouran d'imatges sintètiques.

1.4.6 Comparació dels resultats

Per tal de poder comparar el rendiment del nou model de la xarxa neuronal respecte el anterior, s'ha, de trobar en primer lloc algunes mètriques que permetin quantificar el rendiment dels mateixos. Un cop determinades les mètriques adequades, es pot procedir amb la comparació dels resultats utilitzant les dades de test.

2. Estat de l'art

La literatura en l'àmbit de la segmentació de peixos en un entorn controlat no és gaire extensa. En aquest context, s'ha presentat un mètode basat en el gradient del canal de saturació (Prados R. *et al.*, 2017) pensat específicament per a la seva aplicació sobre imatges capturades pel sistema Deep Vision. Aquest mètode permet segmentar de manera precisa els objectes que es troben en el primer pla respecte el fons, però no és capaç de separar instàncies de diferents peixos en cas de superposició entre espècimens. En aquest projecte es busca ampliar la *pipeline* que defineix el mètode afegint l'ús de Deep Learning per poder aplicar una segmentació d'instàncies. Es donaran més detalls d'aquesta etapa posteriorment.

A l'hora de triar un mètode d'aprenentatge profund cal conèixer les diferències entre diferents capacitats dels mètodes de Deep Learning: classificació d'imatges, detecció d'objectes, segmentació semàntica i segmentació d'instàncies, mostrades a la *Figura 4*.

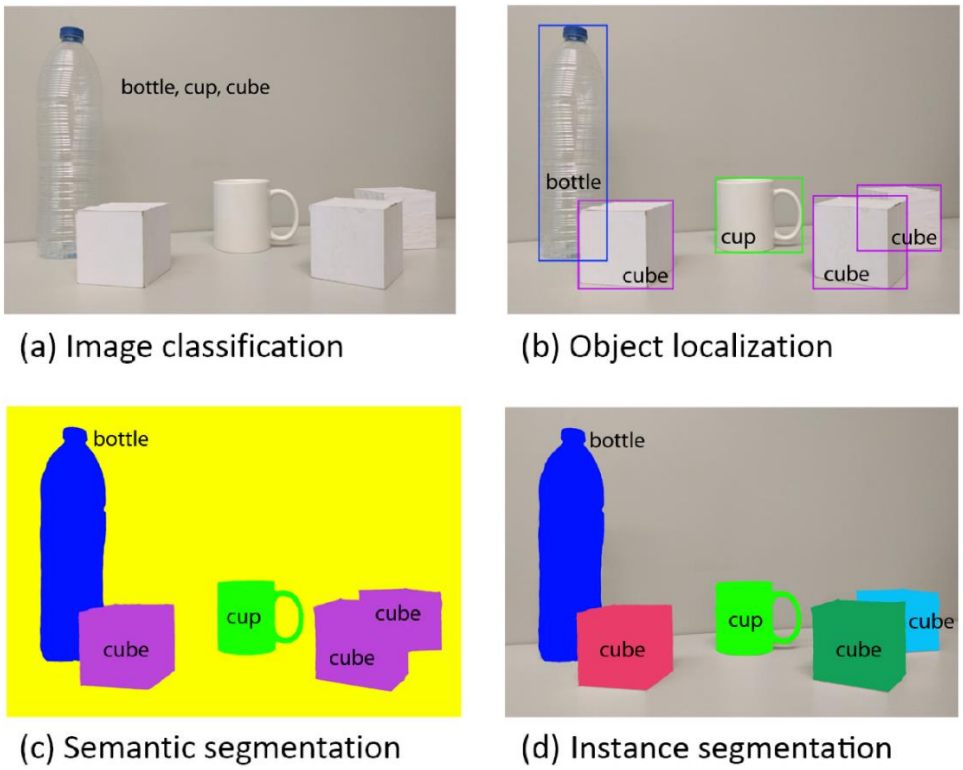


Figura 4. (a) Classificació d'imatges. (b) Detecció d'objectes. (c) Segmentació semàntica. (d) Segmentació d'instàncies (Garcia-Garcia, A. et al, 2017).

- **Classificació d'imatges**. Es busca indicar amb etiquetes els tipus d'objectes que apareixen a la imatge.

- **Detecció d'objectes**. Es busca un requadre que englobi cada objecte, anomenat *bounding box*, al que se li dona una etiqueta.

- **Segmentació semàntica**: Es busca associar cada píxel de la imatge a una etiqueta.

- **Segmentació d'instàncies**: Es busca generar una màscara a nivell de píxel que representi cada objecte per separat al que s'afegeix una etiqueta que indica el tipus d'objecte.

Per poder segmentar per separat i identificar els peixos superposats es necessita un mètode que apliqui una segmentació d'instàncies, i un mètode d'aprenentatge profund que aplica aquest tipus de segmentació és el Mask R-CNN proposat per (Kaiming He *et al.*, 2017).

Per entendre com funciona el Mask R-CNN s'ha d'entendre com funciona l'arquitectura del Faster R-CNN (*Figura 5*), un detector d'objectes basat en aprenentatge profund introduït per (Girshick R. *et al.*, 2015). El procés que segueix aquest algorisme per detectar objectes en una imatge és el següent:

- Extracció d'un mapa de característiques de la imatge utilitzant filtres convolucionals.

- Aplicació de l'algorisme RPN (Region Proposal Network) sobre el mapa de característiques per identificar regions que tenen una probabilitat alta de contenir un objecte.

- Aplicació del *ROI pooling*, en la que s'extreu per cada regió proposada en el pas previ el seu corresponent mapa de característiques.

- Utilització de dues capes de la xarxa totalment connectades (o *fully connected*), que actuen com un classificador, sobre les regions proposades per obtenir de cada una el tipus d'objecte que hi apareix, i un requadre, o *bounding box*, que l'enquadra.

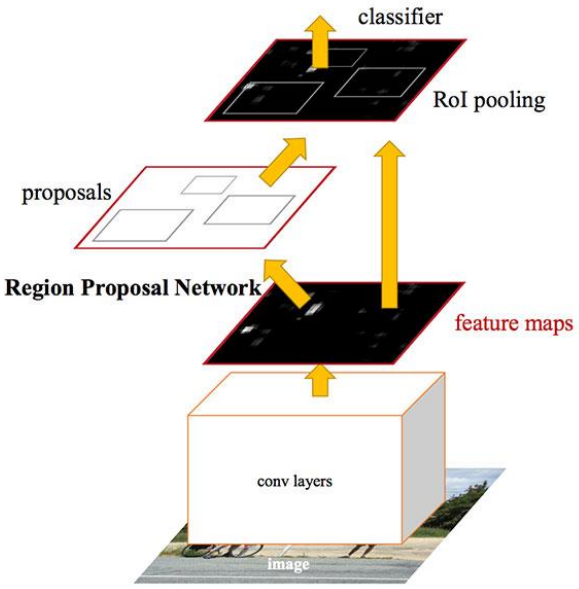


Figura 5. Arquitectura del Faster R-CNN (Girshick R. et al, 2015).

El Mask R-CNN afegeix dos canvis representatius (*Figura 6*) sobre l'arquitectura del Faster R-CNN.

- Canvia el *ROI pooling* per un algorisme anomenat *ROI align*, que també extreu el mapa de característiques de cada regió proposada però de manera més acurada.

- Afegeix a l'arquitectura una branca que surt del procés de *ROI align* i que conté capes convolucionals. Aquesta branca genera una màscara per cadascuna de les regions proposades.

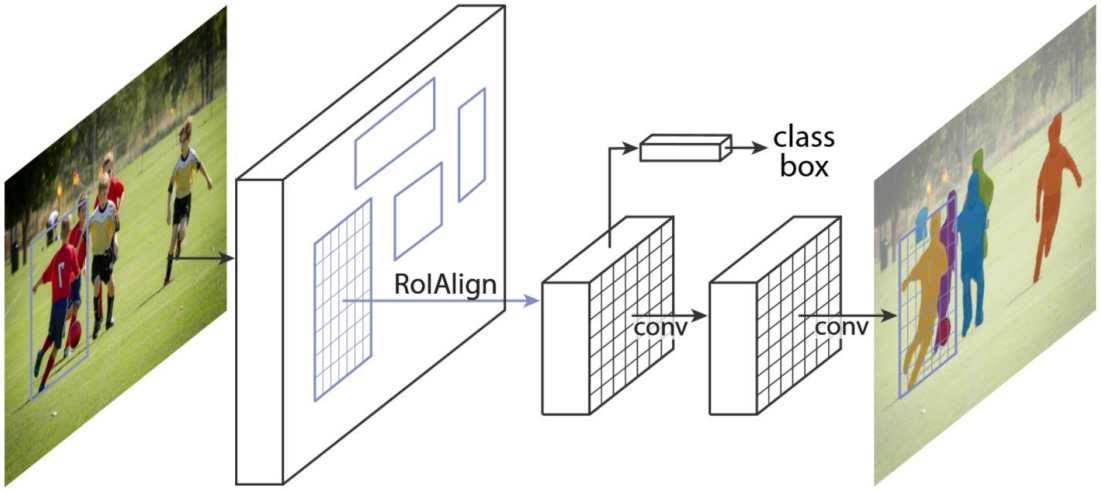


Figura 6. Representació gràfica del procediment aplicat per el Mask R-CNN sobre una regió proposada per la RPN. (a) Imatge que representa una regió proposada per la RPN. (b) *ROI align* que extreu el mapa de característiques de la regió proposada. (c) Alimentació del mapa de característiques a la branca de capes convolucionals per generar la màscara i per altre banda la generació d'una predicció de l'etiqueta de l'objecte (*class box*) i el seu *bounding box*. (d) Imatge segmentada resultat de les dos capes convolucionals. *(imatge provinent de (Kaiming He et al., 2017))*

En resum, la informació que retorna el Mask R-CNN de cada regió proposada d'una imatge conté l'etiqueta que el representa (tipus d'objecte), el *bounding box* que l'enquadra, i una màscara a nivell de píxels que indiquen quins pertanyen a l'objecte, fet que posiciona aquest algorisme com a un perfecte candidat per aplicar la segmentació d'instàncies sobre les imatges capturades amb el sistema Deep Vision.

3. Planificació

Per poder assolir el propòsit del projecte s'ha definit una *pipeline*, que es mostra a la *Figura 7*. En aquesta *pipeline* es volen automatitzar en la major mesura possible els seus processos, per tal que es puguin aplicar a un major nombre d'imatges amb el mínim d'interacció humana.

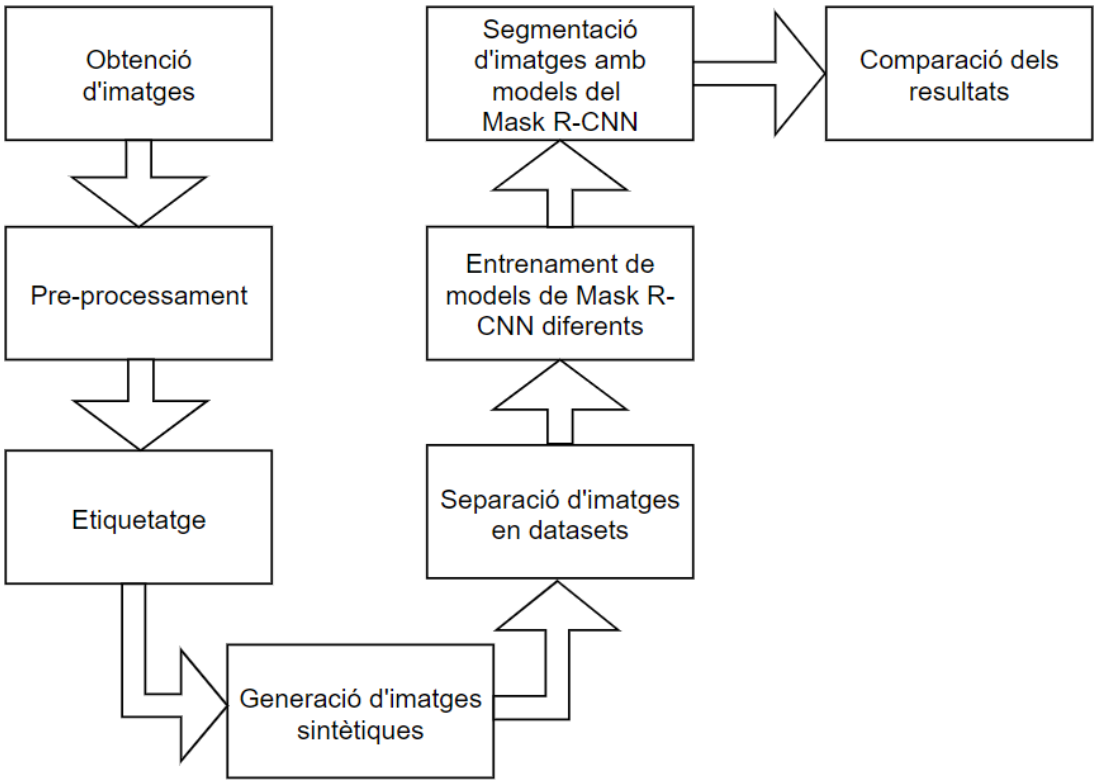


Figura 7. Esquema de la *pipeline* de treball

4. Obtenció i selecció de dades

En tractar-se d'un projecte de recerca encarat a millorar la segmentació dels peixos existents en imatges obtingudes pel sistema Deep Vision, el principal component necessari és el de tenir un data set amb imatges obtingudes pel mateix.

Actualment es disposa d'una gran quantitat d'imatges (*Figura 8*) que s'han obtingut en diferents expedicions realitzades al Mar del Nord, però en aquest projecte, per qüestions de temps, s'ha limitat el treballar a les que corresponen a l'expedició *REDUS Vendla*, a l'hora d'afitar paràmetres existents com poden ser el canvi de color o la diferència en la il·luminació deguts a la utilització de càmeres diferents, tal i com es pot observar a la *Figura 9*. Aquesta expedició es va dur a terme entre el 10 i el 24 de Maig del 2017 vaixell de pesca Vendla en aigües territorials noruegues, i es va utilitzar el sistema Deep Vision per poder monitoritzar els peixos que entraven a la xarxa.

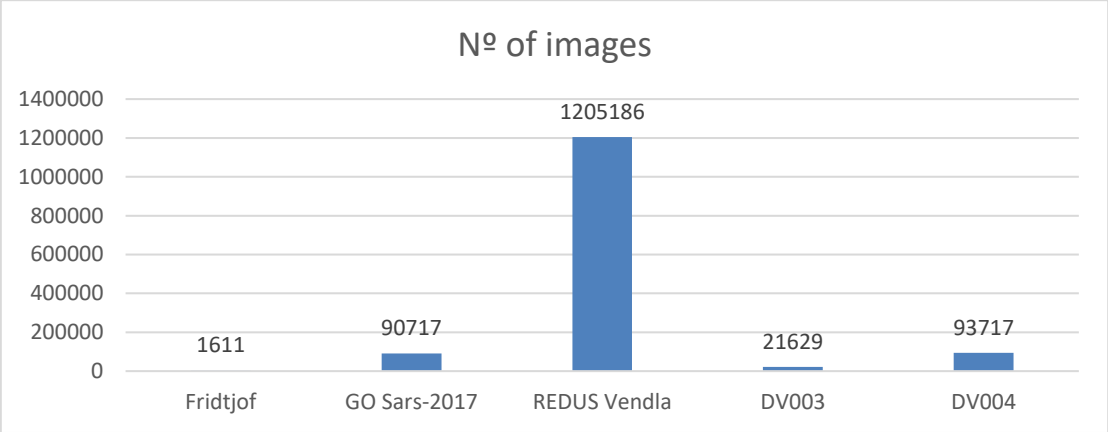
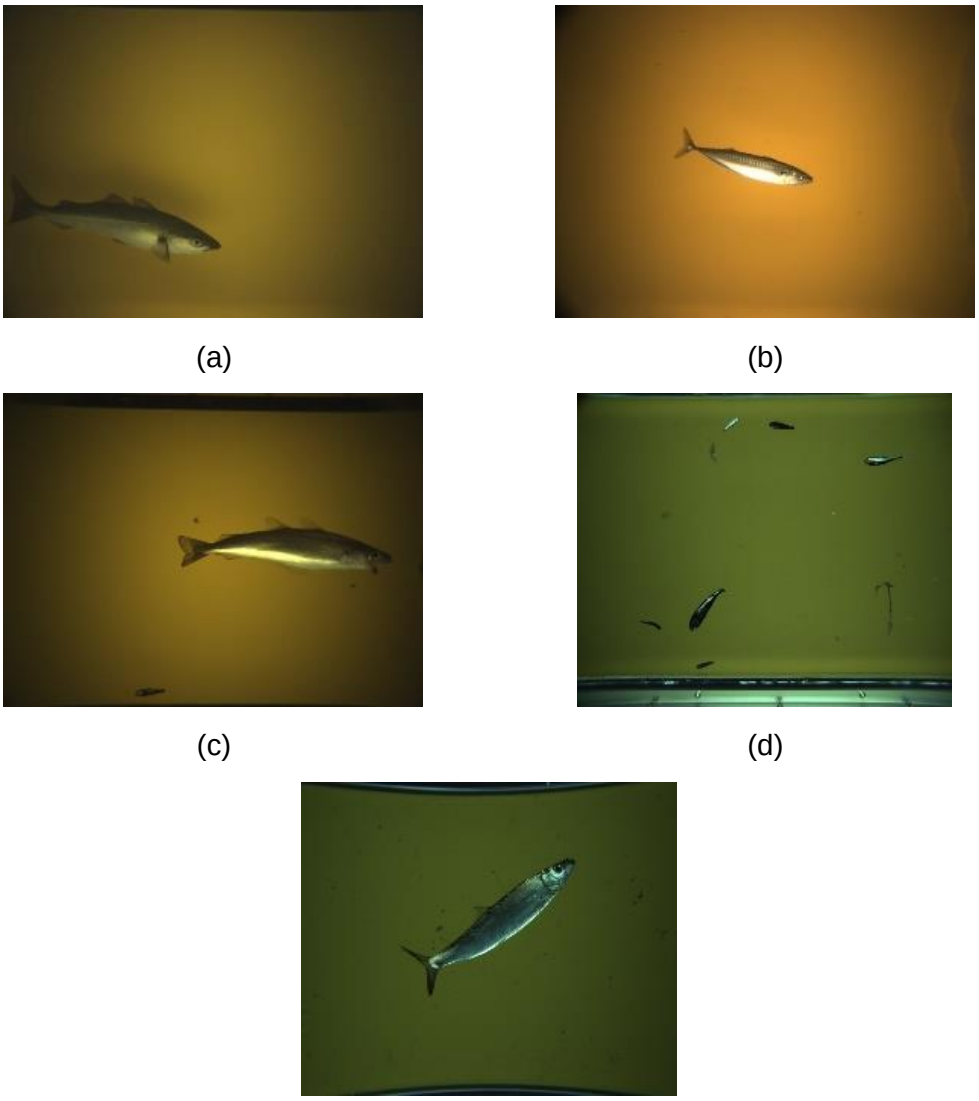


Figura 8. Gràfica que representa el numero d'imatges que contenen els diferents datasets amb imatges obtingudes amb el sistema Deep Vision.



(a) (b)

(c) (d)

(e)

Figura 9. Imatges representatives dels diferents datasets disponibles. (a) Dataset Fridtjof Jansen. (b) Dataset GO Sars. (c) Dataset Redus Vendla. (d) Dataset DV003. (e) Dataset DV004.

A partir d'ara, es referirà a aquest dataset amb el nom *RVendla*. Les imatges d'aquest dataset representen diferents episodis que es produeixen durant la pesca, com es mostra a la *Figura 10*. En aquest projecte s'ha cercat fer una primera classificació de les imatges entre les que són buides o tenen peixos en primer pla.

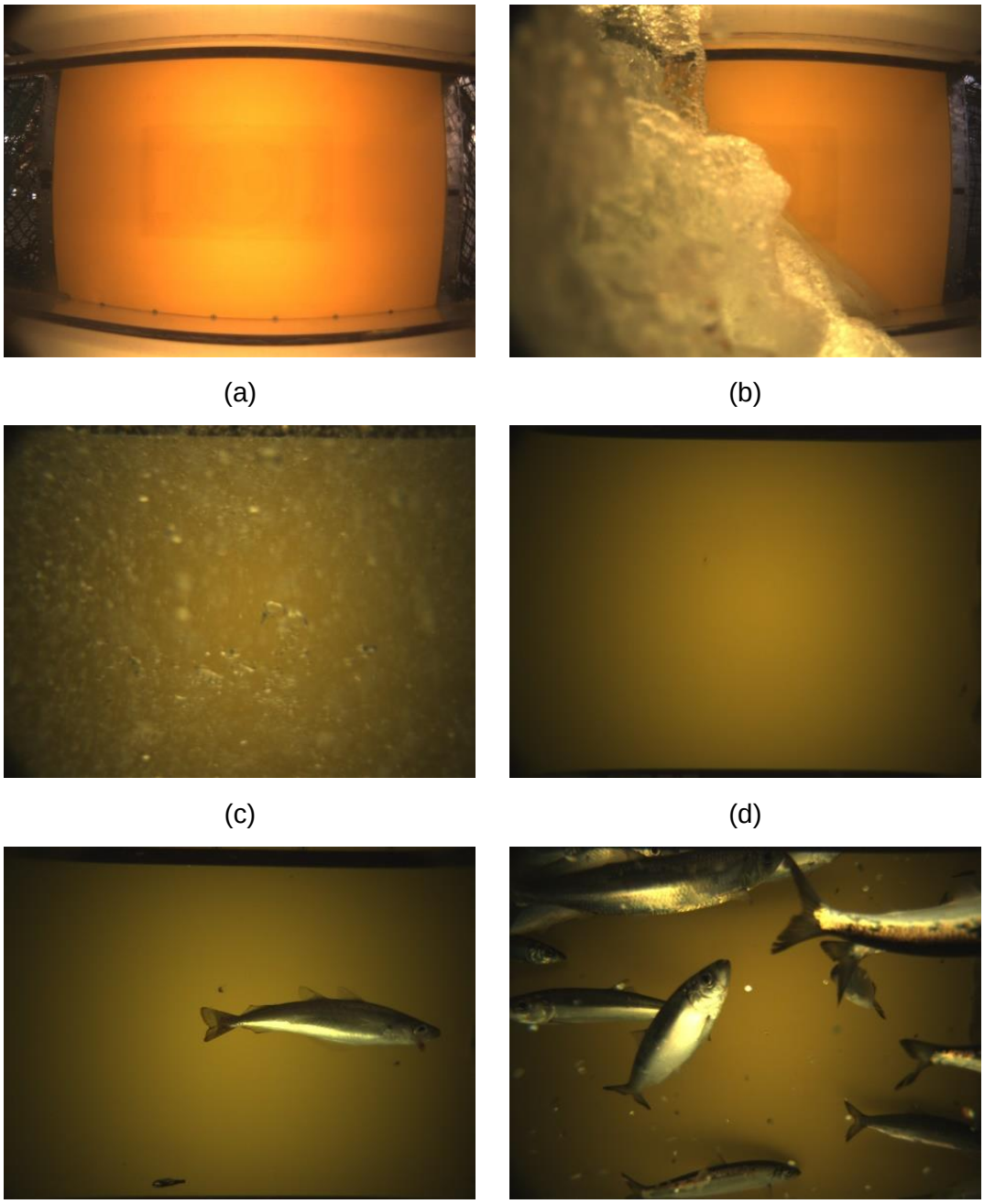


Figura 10. Diferents episodis durant la pesca capturats amb el sistema Deep Vision. (a) Sistema fora de l'aigua. (b) Sistema entrant a l'aigua o sortint. (c) Sistema dins de l'aigua però amb bombolles provinents de la immersió. (d) Sistema completament submergit sense cap peix. (e) Apareguts els primers peixos capturats per la xarxa. (f) Xarxa molt concorreguda indicant la captura d'un banc de peixos.

Per poder classificar les imatges s'ha utilitzat un procés anomenat *background subtraction*, que permet detectar si hi ha objectes en primer pla. Aquest procés és molt utilitzat per detectar objectes que es mouen en seqüències d'imatges a les que es sap que les càmeres són estàtiques i que el fons no canvia, com és el cas del sistema Deep Vision. Per detectar els objectes del primer pla es calcula la diferència entre la imatge actual i una imatge de referència (també anomenat *background*), es converteix la matriu de diferències en una imatge binària, donat un valor de llindar i, el resultat, representa amb l'estat *cert* els píxels que s'estima que pertanyen a un objecte. Les etapes d'aquest procés es mostren a la Figura 11.

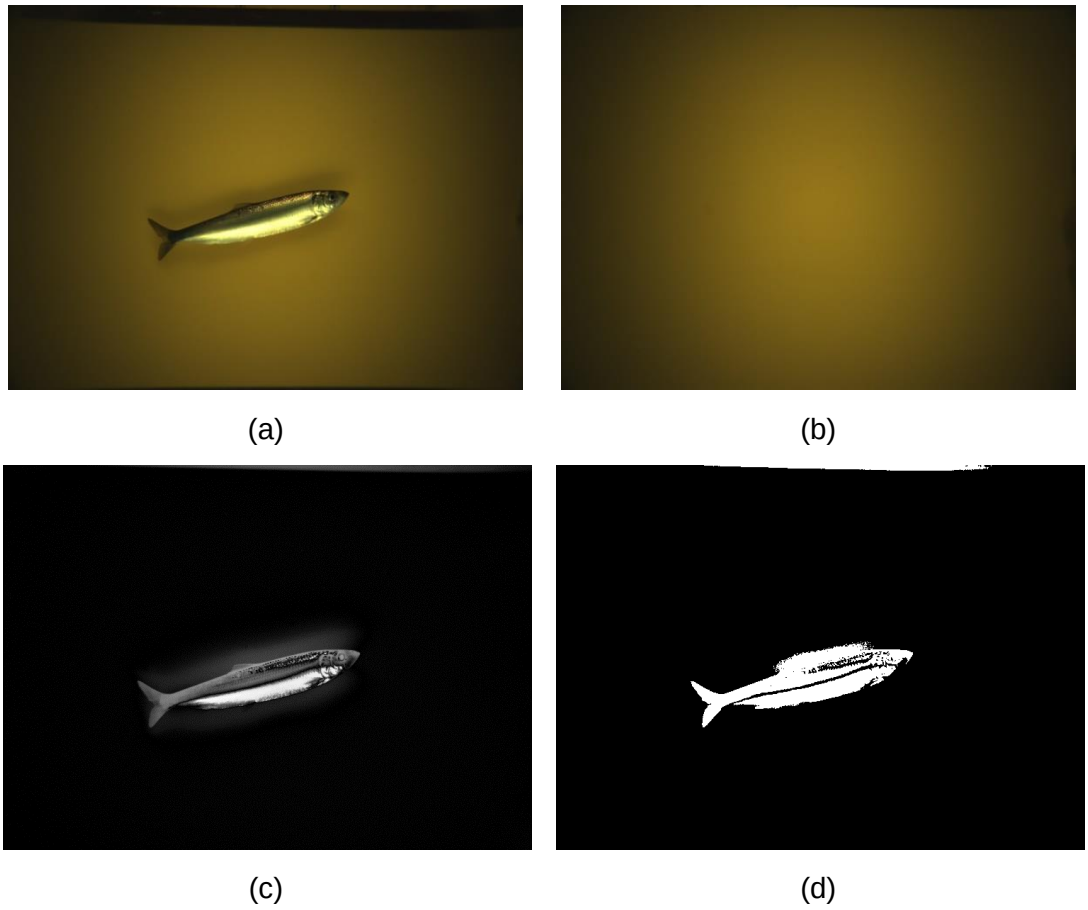
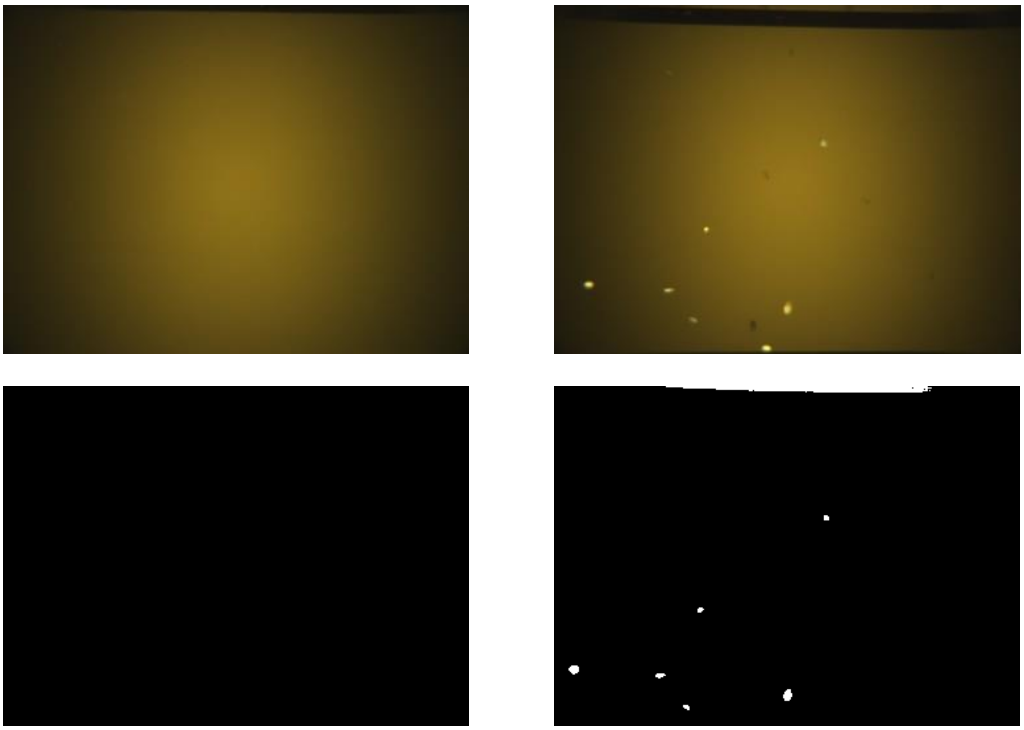


Figura 11. *Background subtraction*. (a) Imatge d'entrada. (b) Imatge de fons. (c) Matriu resultant de calcular les diferències entre (a) i (b) després de ser convertides a nivell de gris. (d) Resultat transformat a una imatge binària utilitzant un llindar de valor 30. La suma dels píxels d'aquesta imatge binària diferents de zero, és la que s'utilitza per classificar la imatge d'entrada.

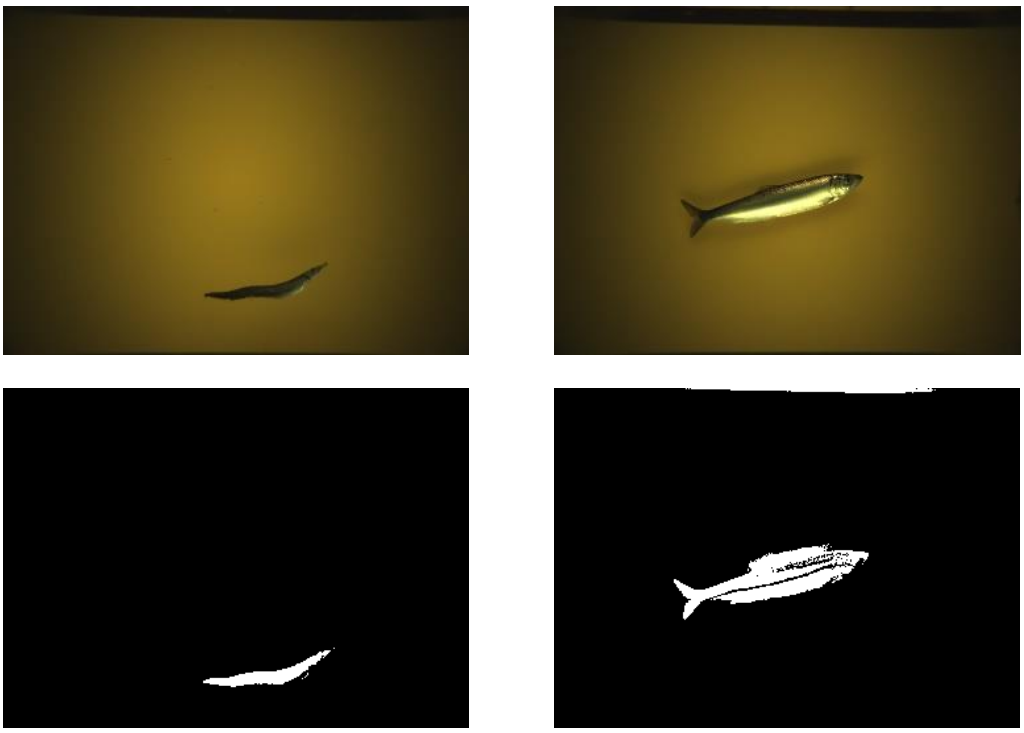
La majoria d'imatges sense peixos contenen partícules, raó per la qual no s'ha utilitzat una única imatge com a *background*, i s'ha calculat una imatge nova a partir d'un conjunt, donant lloc a un resultat més robust. Cada píxel de la nova imatge de fons generada representa la mediana dels valors del mateix píxel de les imatges del conjunt.

Per classificar les imatges entre buides o amb objectes, s'ha quantificat el número de píxels que representen objectes en primer pla que s'ha obtingut aplicant *background subtraction* i, en cas de superar un valor v donat, s'ha considerat de que es tractava d'una imatge que conté peixos. Aquest valor v s'ha trobat de forma empírica, calculant la suma dels píxels sobre imatges diferents amb instàncies de peixos petits o gambes, i fent una estimació sobre el llindar. Alguns dels resultats utilitzats per l'estimació de v es mostren a la *Figura 12*. No s'ha fet un estudi exhaustiu per fer aquesta classificació el més acurada possible, ja que es disposa d'un gran nombre d'imatges del dataset *RVendla*, fet que permet ignorar alguna imatge que contingui peixos, i pel fet de que posteriorment les imatges són processades, mantenint la possibilitat d'eliminar aquelles que s'han classificat incorrectament durant la selecció d'imatges sense cap peix en primer pla.



$\sum píxels > 0 = 0$ (a)

$\sum píxels > 0 = 12338$ (b)



$\sum píxels > 0 = 9899$ (c)

$\sum píxels > 0 = 43890$ (d)

Figura 12. Alguns resultats de les imatges binàries i les sumes dels píxels al aplicar *background subtraction* que s'han utilitzat per estimar el valor v. (a) Imatge que no presenta cap partícula. (b) Imatge amb partícules. (c) Imatge amb un peix de petites dimensions. (d) Imatge amb un peix. En les imatges (b) i (d) no s'ha aplicat cap preprocessat per eliminar la barra superior.

Als resultats de *Figura 12* (b) i (d) es pot veure com també s'ha inclòs la franja que apareix a la part superior de la imatge com a objecte, incrementant considerablement la suma de píxels. La inclusió d'aquests píxels és errònia ja que es tracta d'un element del fons. Per evitar incloure aquesta franja, abans d'aplicar el *background subtraction*, es defineix sobre quina part de les imatges (mitjançant una regió d'interès, o ROI) s'aplicarà com es mostra a la *Figura 13*.

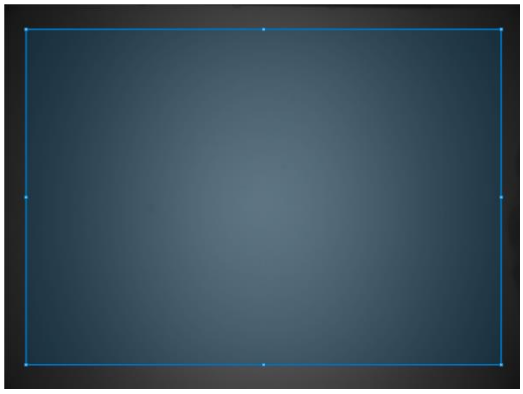


Figura 13. Captura del procés de definició de la zona de les imatges sobre la que s'aplicarà el *background subtraction.*

El valor v que s'ha definit com a llindar per a la classificació és $v = 5000$ i a la *Figura 14* es mostra el resultat de la classificació utilitzant-lo. Aquesta selecció permet accedir posteriorment de forma fàcil al conjunt d'imatges que inclouen peixos, i que són, en la seva gran majoria, les que s'han fet servir en etapes posteriors del projecte.

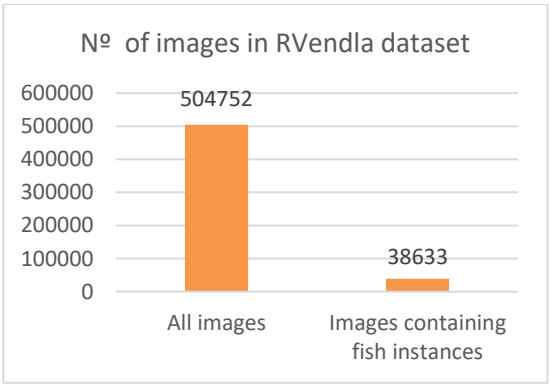


Figura 14. Representació gràfica de la quantitat d'imatges que conté el dataset *RVendla* en total (esquerra) i de les imatges classificades com imatges que contenen instàncies de peixos.

5. Preprocessat

5.1 Correcció de la il·luminació no uniforme

La il·luminació del sistema Deep Vision està formada per dues tires de LEDs potents situats a dalt i a baix de l'estructura, però que no apunten de forma directa a l'escena per tal d'evitar reflexions especulars sobre les escames dels peixos. Aquest sistema però, presenta un degradat en la il·luminació, que s'accentua a major distància del centre d'aquestes, tal com es mostra a la *Figura 15*. Aquest efecte es coneix com vinyetatge (o *vignetting*, en anglés), i s'ha volgut corregir per tal de mantenir la il·luminació uniforme i facilitar el processat posterior de les imatges.

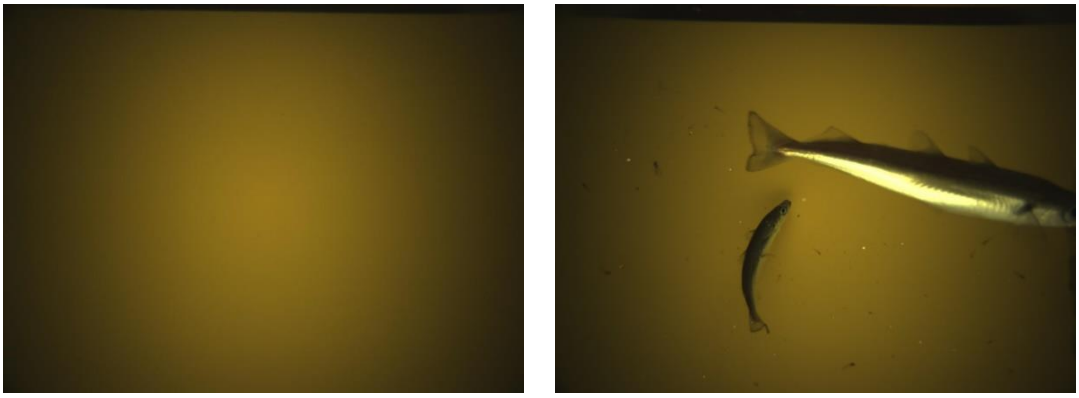
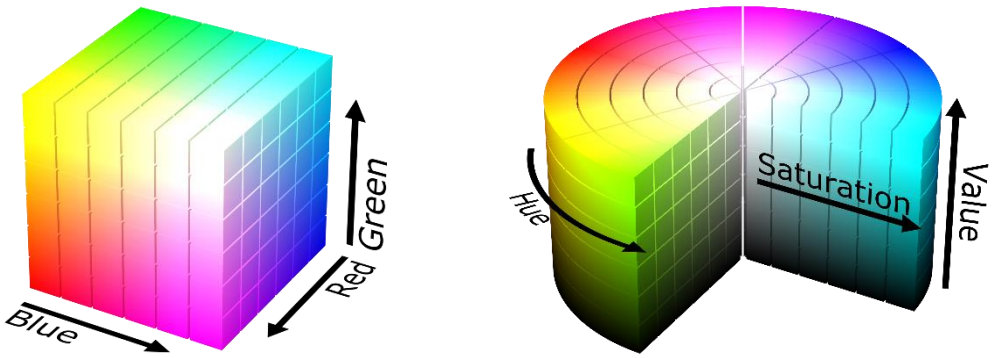


Figura 15. Efecte vinyetatge en imatges del dataset REDUS Vendla. La intensitat de la il·luminació va disminuint a mesura que s'allunya del centre de la imatge.

Per corregir aquesta no-uniformitat en la il·luminació en les imatges, s'ha utilitzat el coneixement de la uniformitat real del fons de l'estructura del sistema Deep Vision, i el fet de que la il·luminació es manté constant a totes les imatges. Sabent que el fons real és de color groc i totalment uniforme (dissenyat així per augmentar el contrast amb els peixos, tenint present que no existeix cap peix d'aquest color a l'Atlàntic), s'ha optat per agafar una imatge sense cap objecte en el primer pla, utilitzar-la per calcular les diferències d'intensitats a nivell de píxel, i generar un patró que inclogui tota aquesta informació. Aquest patró es pot utilitzar posteriorment sobre altres imatges per compensar les diferències d'intensitats dels diferents píxels.

Les imatges del datasets estan emmagatzemades en el model de colors RGB i, per per tal d'evitar possible pèrdues de correlació entre els diferents canals, s'ha optat per transformar les imatges al model HSV (de l'anglès *Hue*, *Saturation*, *Value*, és a dir, Tonalitat, Saturació, Valor). Aquest model de color permet treballar millor amb les intensitats de cada píxel ja que les variacions en la il·luminació només afecten al component *Value* sense dependre dels components to i saturació, tal i com es pot observar en la *Figura 16*.



(a) (b)

Figura 16. Representacions de models de color. (a) Model de color RGB (*Red* - Vermell, *Green* - Verd, *Blue* – Blau). (b) Model de color HSV (*Hue* - to, *Saturation* - Saturació, *Value* - Intensitat).

Amb les imatges transformades a l'espai HSV, el següent pas consisteix en d'agafar un valor d'intensitat com a referència i calcular, per cada píxel, la diferència del píxel actual respecte d'aquest. Amb la informació de les diferències d'intensitats de cada píxel, es pot crear un patró que pot ser utilitzat posteriorment per compensar les intensitats dels píxels en altres imatges generades en el mateix entorn controlat del sistema d'obtenció d'imatges.

Per generar un patró robust, s'han utilitzat unes desenes d'imatges a l'hora de fer el càlcul. La fórmula que representa el càlcul d'aquest patró per quantificar variació en la il·luminació és la que es mostra a continuació:

$$P(x,y) = \frac{\sum_i \frac{m_i}{V_i(x,y)}}{g}$$

On ***i*** indica la instància de la imatges utilitzada per calcular el patró i $\boldsymbol{V_i}$ representa el component *Intensitat* de cada una d'elles, ***g*** és el número d'imatges que s'utilitzen i ***m*** és la mitjana dels valors d'intensitats de la imatge actual.

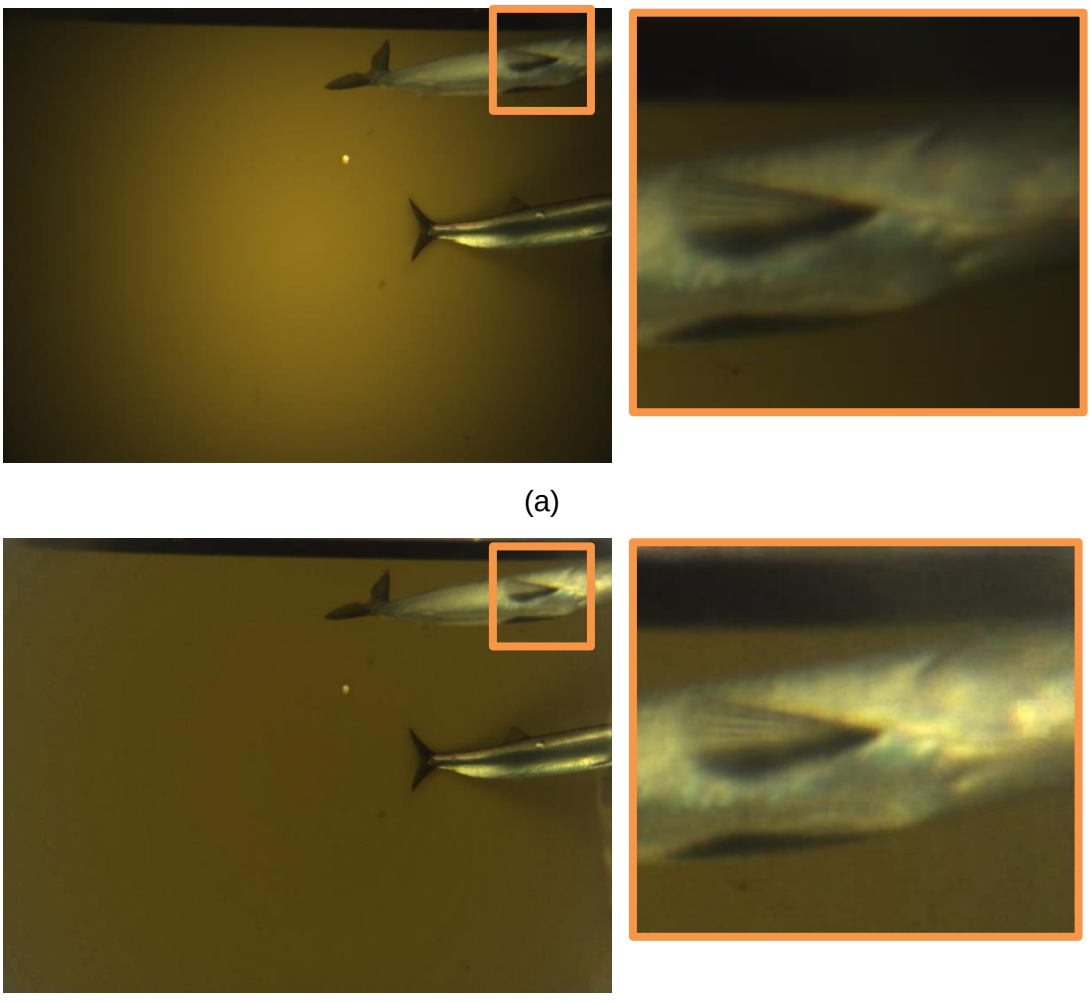
El patró resultant representa un conjunt de pesos que, al multiplicar-lo pel component d'intensitat d'una altra imatge obtinguda en el mateix entorn controlat, en la que s'han obtingut les imatges utilitzades per calcular el patró, compensa la diferència d'il·luminació eliminant el efecte vinyetatge, com es pot observar a la *Figura 17*.

(a)

(b)

Figura 17. Correcció de la il·luminació no-uniforme. Imatge resultant (b) després de compensar la il·luminació de la imatge (a), utilitzant un patró calculat amb imatges capturades prèviament amb el mateix sistema.

Un factor que s'ha de tenir en compte a l'hora de fer la correcció de la il·luminació és que a les parts més fosques de la imatge inicial, hi ha poca informació sobre el color real (degut a la falta de sensibilitat de la càmera i a l'emmagatzematge de les mateixes en formats amb compressió amb pèrdua) i normalment presenta soroll. En la imatge original aquesta falta d'informació no és visible, però en el moment d'aplicar la correcció incrementant les intensitats d'aquestes zones, també s'incrementa el soroll, com es pot observar a la *Figura 18*.



(a)

(b)

Figura 18. Soroll amplificat en les zones més fosques de la imatge al aplicar la correcció de la il·luminació no-uniforme. (a) Imatge original. (b) Imatge corregida.

6. Etiquetatge

L'etiquetatge d'imatges de forma manual és un procés lent i costós. S'ha de tenir en compte que per entrenar una xarxa neuronal és crucial disposar d'una quantitat elevada d'imatges etiquetades correctament i que representin de manera més completa el domini sobre el qual s'aplica. En aquest projecte es busca solucionar el problema de disposar de poca quantitat d'imatges etiquetades mitjançant la generació d'imatges sintètiques, però per generar aquestes de forma àgil també cal disposar d'imatges ja etiquetades. A més a més, la quantitat disponible d'instàncies de peixos etiquetats afecta directament a la varietat de les imatges que es generen amb elles.

Posant com a exemple que es disposés d'un dataset que només contingués vint peixos segmentats, i es generessin vint-mil imatges sintètiques amb aquests, encara que s'apliquessin transformacions, la informació redundant que donarien aquestes imatges durant l'entrenament a la xarxa seria mínima respecte a la gran quantitat de tipus de peixos, mides i posicions que poden arribar a adoptar a la realitat.

Degut a això, s'ha buscat una forma d'etiquetar imatges de forma precisa i en el que es necessiti el mínim d'interacció humana, per tal de reduir el temps i el treball necessari per generar un conjunt de dades prou gran per donar varietat a les imatges generades.

El primer pas per etiquetar és estudiar de quina forma es vol guardar la informació sobre les regions. Hi han formes diferents de guardar les màscares obtingudes a partir de aplicar segmentacions, com pot ser l'ús de polígons en que cada regió es guarda com un conjunt de punts que pertanyen als vèrtex del mateix, que el descriuen, o la utilització de màscares de la mateixa mida que la imatge segmentada i amb una sola dimensió en la que cada píxel agafa el número de la regió al que pertoca, com es mostra a la *Figura 19*.

(a) (b)

Figura 19. Representació de l'emmagatzematge de la informació de les etiquetes generades sobre una imatge en una màscara. (a) Imatge original. (b) Màscara en escala de gris en que el valor de cada píxel indica a quina regió pertany. El valor 0 és l'utilitzat per representar el fons.

En aquest projecte, la part més restringida del *pipeline* és l'algorisme del Mask R-CNN ja que s'ha utilitzat una implementació ja existent. La implementació del Mask R-CNN, aplicant alguna modificació al codi principal, permet passar la informació de les etiquetes de les dues formes descrites anteriorment, utilitzant polígons per descriure les regions o utilitzant una màscara que indiqui a quina regió pertany cada píxel. S'ha optat per la segona opció ja que aquest tipus de màscares es poden tractar com a matrius directament, cosa que suposa una facilitat a la hora de preprocessar-les amb MATLAB®, el software utilitzat per implementar els scripts utilitzats. També s'ha decidit utilitzar el valor 0 per representar el fons de les imatges.

Al no dependre d'un temps limitat per fer el processat, s'ha optat per la utilització del procés de refinament de màscares proposat en l'article de segmentació automàtica (Garcia *et al.*, 2019) on s'utilitza un model de la xarxa neuronal Mask R-CNN ja entrenat sobre imatges obtingudes del sistema Deep Vision i un algorisme de segmentació basat en el gradient, per generar un etiquetatge precís. Finalment s'ha utilitzat una eina implementada amb la plataforma MATLAB® per comprovar i corregir les imatges generades pel procés prèviament descrit.

L’algorisme de segmentació de basat en el gradient permet trobar una segmentació precisa, però no distingeix entre dos peixos diferents si aquests es troben en contacte, tractant-los com a una sola regió i donant lloc a una sotasegmentació, com s’observa a la *Figura 20 (b)*. Per solucionar-ho s’utilitza la informació que dóna la segmentació imprecisa resultant del model antic del Mask R-CNN (*Figura 20 (c))* per intentar separar correctament cada instància del peix mantenint la precisió de la màscara del primer algorisme.

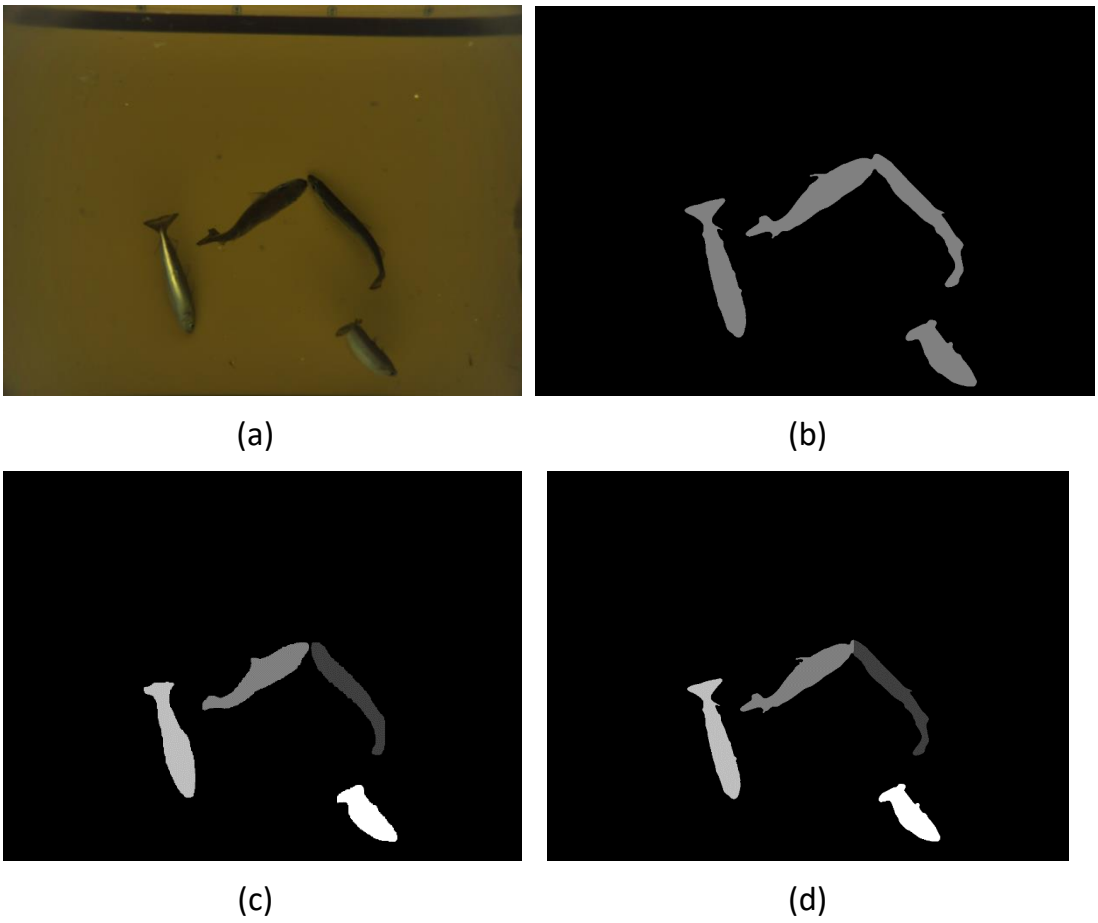


Figura 20. Imatges implicades durant el refinament. (a) Imatge original. (b) Màscara generada amb l’algorisme de segmentació basat en el gradient. (c) Màscara generada amb el Mask R-CNN. (d) Resultat del refinament.

6.1 Segmentació amb Mask R-CNN

El Mask R-CNN genera la segmentació sobre una imatge de baixa resolució que s'obté aplicant un procés de *downsampling* a la imatge original, i que després es torna a augmentar fins la grandària inicial. La suma d'aquest *downsampling* i de la utilització d'un model de la xarxa neuronal entrenada amb poques imatges equival a una segmentació amb un contorn imprecís i en el major dels casos de menor àrea en comparació a la instància del peix, com es pot veure a la *Figura 21*.

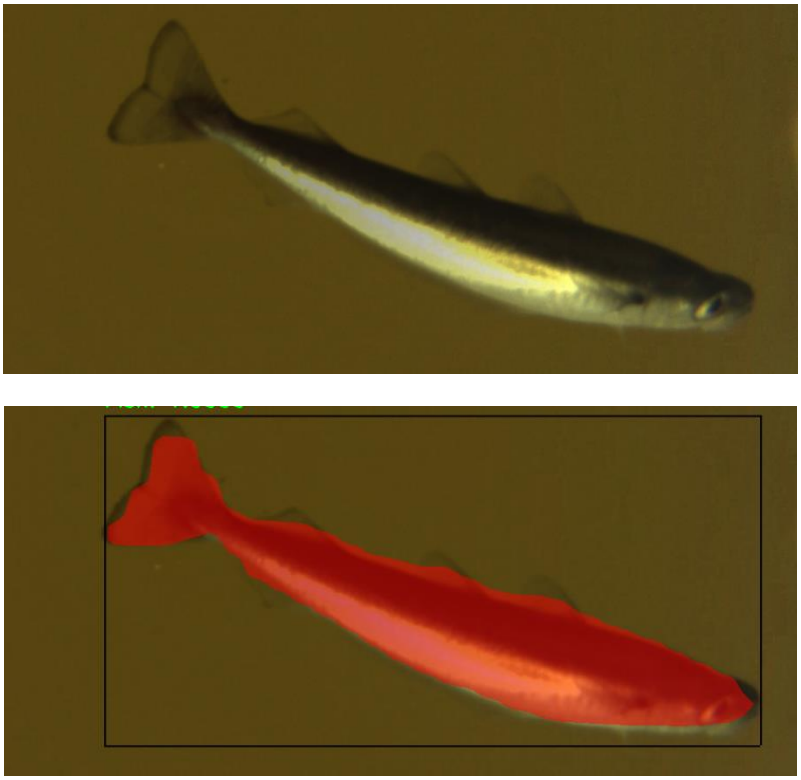


Figura 21. Imatge que representa la imprecisió de la segmentació del Mask R-CNN utilitzant el primer model disponible.

6.2 Segmentació basada en el gradient

L'algorisme de segmentació basat en gradient explicat en l'article (Prados R. *et al.*, 2017) utilitza una imatge de fons o *background* a més de la imatge a segmentar. El procediment que segueix aquest mètode, mostrat a la *Figura 22*, utilitza el canal de saturació de les imatges transformades a l'espai de color HSV de la imatge original i del *background* per calcular el gradient i utilitzar aquesta informació binaritzada per fer una primera segmentació del peix. La imatge es retalla fins tenir una regió que englobi amb poc marge aquesta segmentació i, amb la ajuda d'un segon càlcul de gradients sobre la imatge original i el background s'acaba de completar la màscara que representa el peix.

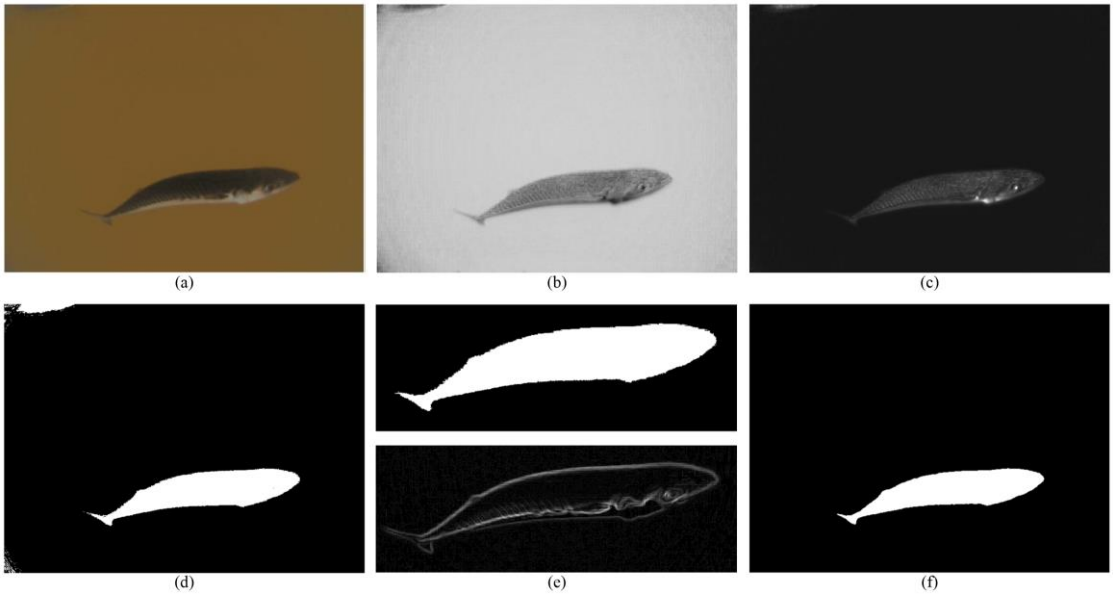


Figura 22. Imatges extretes del article (Prados R. *et al.*, 2017) que descriuen el procés de segmentació basat en el gradient. (a) Imatge linealitzada amb la il·luminació no-uniforma corregida. (b) Canal de saturació de la imatge representada en el model de colors HSV. (c) Diferència d'intensitat entre la imatge original i *background*. (d) Matriu de diferències d'intensitat passada a binari aplicant un llindar. (e) Àrea extreta que conté un peix de d'aplicar el anàlisis de regions (dalt) i els gradients de la imatge original existents dins de l'àrea extreta (baix). (f) Màscara final segmentada que representa el peix.

6.3 Procés de refinament

El *pipeline* que segueix el procés de refinament està representat a la *Figura 23*. Utilitzant les dues màscares obtingudes amb els algorismes descrits prèviament, s'utilitza la segmentació del Mask R-CNN i s'aplica la operació morfològica d'un cert número de píxels sobre la regió de cada peix, com es mostra a la *Figura 24*. Seguidament, es busca, per a cada regió de la nova màscara, els píxels que pertanyen a la màscara obtinguda per l'algorisme basat en gradient, donant lloc a una nova màscara que combina la detecció de diferents peixos amb el Mask R-CNN i de la precisió de l'algorisme basat en el gradient.

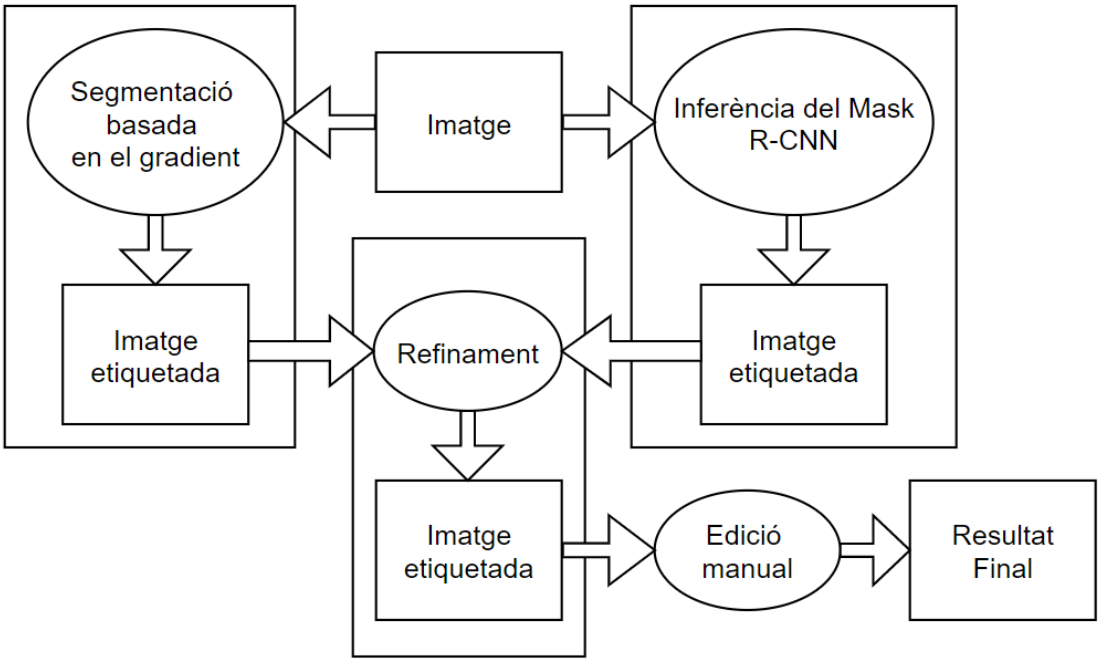


Figura 23. *Pipeline* que segueix el procés de refinament. Partint d'una imatge es generen una màscara amb l'algorisme basat en el gradient i una màscara amb el model del Mask R-CNN. La informació de les dues màscares es combinen en l'algorisme de refinament. La màscara resultant es processa manualment per generar una altre màscara que representa el *groundtruth* de la imatge.

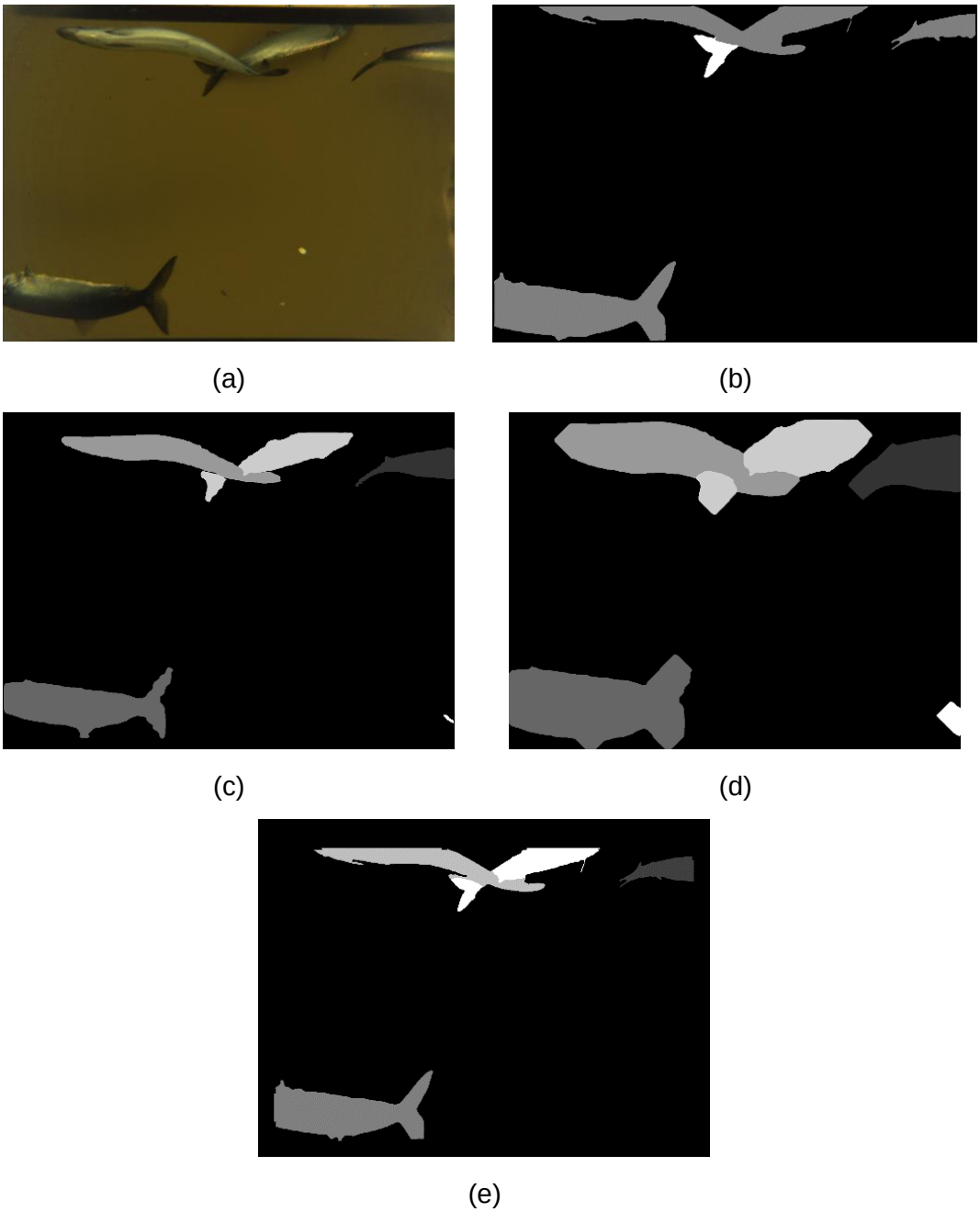


Figura 24. Procés del refinament. (a) Imatge original. (b) Màscara generada amb l'algorisme de segmentació basat en el gradient. (c) Màscara generada amb el Mask R-CNN. (d) Aplicació de la operació *dilate* sobre la màscara generada per el Mask R-CNN. (e) Resultat final del refinament.

La qualitat de les segmentacions obtingudes aplicant el refinament depèn en gran part dels resultats dels algorismes, i en la major part dels casos presenten defectes. En cas del algorisme basat en el gradient, la precisió depèn d'alguns paràmetres modificables, la majoria dels quals s'han de determinar de forma empírica, i de la quantitat de peixos i ombres que apareixen en les imatges, que pot donar lloc a imatges sobresegmentades. D'altra banda, la màscara obtinguda utilitzant un model de Mask R-CNN també pot presentar segmentacions incorrectes, tenint en compte que només es disposa d'un model entrenat amb poques imatges i poques instàncies de peixos.

Les imatges obtingudes a partir del refinament s'han comprovat i corregit manualment utilitzant una aplicació implementada en MATLAB® durant l'estada en l'empresa a Coronis (*Figura 25*). Aquesta aplicació permet afegir, eliminar i modificar, a nivell de píxel, les regions de les màscares. Permet generar màscares des de zero, però en aquest cas s'ha partit dels resultats del refinament, en els quals bona part de la segmentació és correcta, i redueix el processament que s'ha d'aplicar.

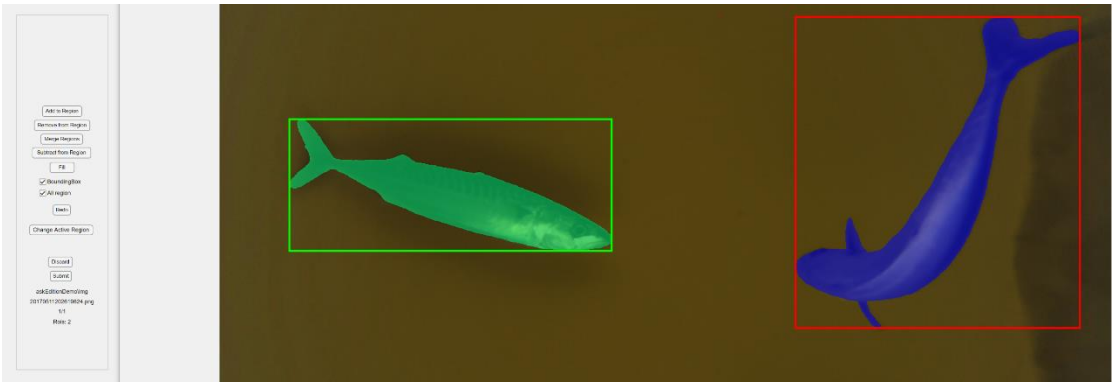


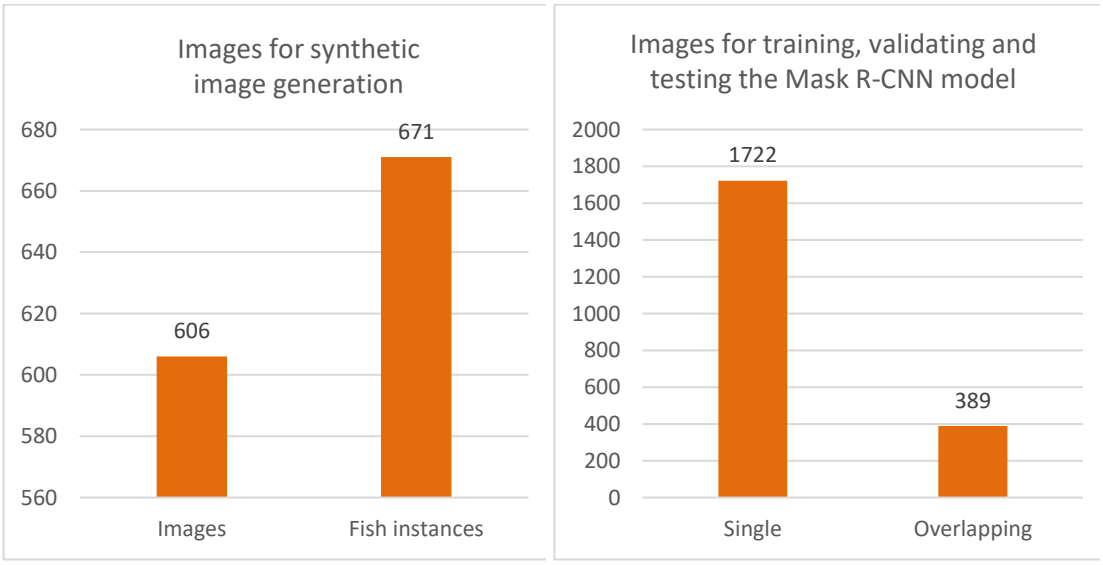
Figura 25. Captura de de l'aplicació implementada amb MATLAB® que permet editar les màscares.

Les etiquetes que s'han generat aplicant el procés de refinament i l'edició manual per algunes persones durant un parell de mesos, s'han separat en dos conjunts:

- Dataset que s'utilitzarà com a material per generar les imatges sintètiques. Imatges que contenen com a mínim una instància completa de peix. Amb instància completa s'entén el peix es veu completament sense cap mena d'oclusió, ja sigui per causa d'un altre peix o per la vora de la imatge.

- Dataset per entrenar i testejar el model de la xarxa neuronal. Dataset que inclou tot tipus d'imatges que contenen peixos, i es subdivideix en dades d'entrenament, validació i test. El conjunt de test o de prova està format, principalment, per imatges amb instàncies que presenten oclusions, per poder quantificar correctament el rendiment sobre aquest tipus d'imatges, que representa la finalitat d'aquest projecte.

La quantitat d'aquestes d'imatges processades i assignades als dos conjunts de dades es pot observar en la taula representada a la *Figura 26*.



(a) (b)

Figura 26. Taula que representa les imatges disponibles en els dos diferents datasets. (a) Quantitat d'imatges i de les instàncies de peixos que hi apareixen del dataset preparat per generar imatges sintètiques. (b) Quantitat d'imatges que inclou el dataset preparat per l'entrenament i test del Mask R-CNN diferenciant entre les que només contenen peixos aïllats i també de superposats.

7. Generació de les dades sintètiques

La correcta generació de les imatges sintètiques és un procediment clau en el projecte. Quan entrenem un model de Machine Learning, el que realment estem fent és ajustar els seus paràmetres de manera que pugui associar una entrada determinada (la nostra imatge) a alguna sortida (una etiqueta, o conjunt d'etiquetes que identifiquen i localitzen els peixos en la imatge). Però les xarxes neuronals d'última generació normalment tenen milions de paràmetres. Lògicament, si tenim molts paràmetres, haurem de mostrar al nostre model de Machine Learning una quantitat proporcional d'exemples per obtenir un bon rendiment, i aquí és on la generació de dades sintètiques ens pot ajudar.

Aquestes imatges han de ser el més semblant possible al model real per no introduir errors en l'aprenentatge de la xarxa neuronal. Amb aquesta idea en ment, s'han estudiat una gran quantitat d'imatges per poder extreure informació sobre les tendències d'aquestes i respondre preguntes com: "En quina posició es troben la gran majoria de peixos?", "Apareixen partícules en la gran majoria d'imatges?", etc.

Un cop s'ha obtingut la idea general sobre el model que han de seguir les imatges generades sintèticament, s'ha estudiat com extreure les instàncies de peixos de les imatges reals i com introduir-les en les noves imatges.

7.2 Extracció

Utilitzant les màscares de les imatges processades a mà i seleccionades per contenir com a mínim un peix sense oclusions, s'ha tingut a l'abast accés directe per poder extreure a nivell de píxels les instàncies dels peixos de cada imatge. Aquest accés directe als píxels de la imatge que pertanyen a les regions diferents i tenint en compte que les imatges es poden tractar com a matrius, permet aplicar modificacions a les instàncies de peixos etiquetades utilitzant una matriu de transformacions planes rígides.

D'altra banda, la diferència de color entre el peix i el fons del sistema Deep Vision és gran. Això provoca que en les vores de les segmentacions dels peixos puguin aparèixer alguns píxels que continguin el color del fons. Al només voler utilitzar els píxels que pertanyen al peix, s'ha decidit aplicar una operació morfològica *erode* o erosionat a les màscares, per tal d'eliminar alguns píxels del contorn, com es pot observar a la *Figura 27*. Aquesta operació pot arribar a eliminar també alguns dels píxels pertanyents al peix, però s'ha decidit que l'objectiu d'eliminar els píxels que pertanyen al fons és més important encara que sigui a costa de perdre part de la informació del peix.

(a)

(b)

Figura 27. Demostració dels píxels de les segmentacions pertanyent al fons de color groc (a) i la seva eliminació aplicant l'operació *erode (b)*.

En aquest cas, les màscares són representades amb una imatge a nivell de gris, que equival a una matriu de dos dimensions a la que cada píxel pot tenir un valor entre 0 i 255. Aplicar l'operació *erode* de la llibreria de MATLAB® sobre aquest tipus d'imatge utilitzant un element d'estructura o un filtre pla, com el que s'ha fet servir en aquest cas (*figura 28*), equival a aplicar un operador local mínim. Explicat de manera diferent, el valor del píxel que s'està processant durant l'erosió agafa el valor mínim dels valors dels píxels que cauen dins de l'element d'estructura, com es mostra a la *figura 28*.

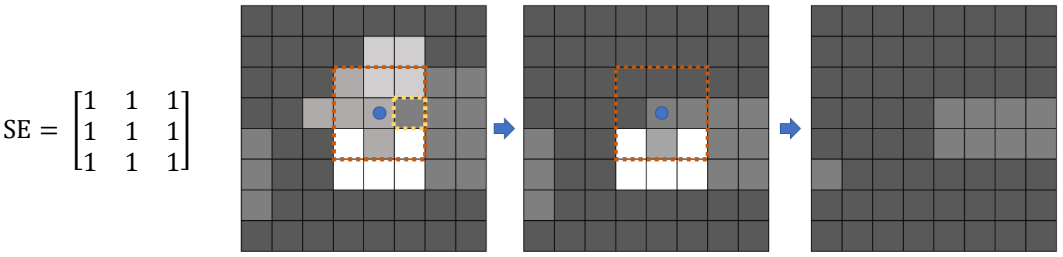


Figura 28. Exemple d'aplicació de l'operació *erode* sobre una imatge a nivell de gris. (a) Representació de l'element d'estructuració. (b) Element de estructuració marcat en taronja aplicat sobre un píxel marcat en color blau. En color groc es representa el píxel amb menor valor dels que es troben dins de l'element d'estructuració. (c) Imatge en processament, en la que es pot veure els valors que s'han canviat abans del píxel processat actual (color blau). (d) Resultat d'aplicar l'*erode* sobre la imatge. El contorn de la imatge es descarta

7.3 Transformació

Per donar més diversitat a les imatges generades sintèticament s'ha decidit aplicar transformacions sobre les instàncies de peixos segmentades. Aquestes transformacions, que es mostren a la *Figura 29,* inclouen translacions, rotacions, escalats i cisallament. Un conjunt d'aquestes transformacions equival a una projecció afí.

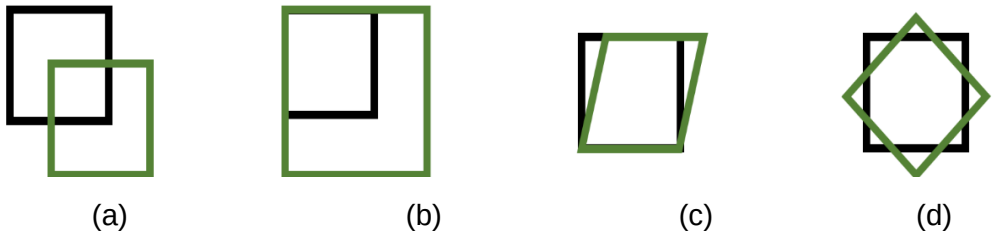


Figura 29. Transformacions en dos dimensions. El quadrat de color negre representa l'estat inicial i el verd la seva transformació final. (a) Translació. (b) Canvi d'escala (c) Cisallament. (d) Rotació. Imatges

Per aplicar aquestes transformacions s'han fet servir coordenades homogènies, que permeten representar les transformacions planes com a matrius 3x3. Aquestes transformacions inclouen des de translacions, escalats i rotacions, fins a transformacions afins i projectives. Aquesta representació, a més, permet encadenar sèries de transformacions fent servir simples multiplicacions de matrius 3x3.

En concret, en aquest treball s'han utilitzat transformacions planes com la homografia que es mostra a continuació (coneguda com a *similarity* a la bibliografia (Hartley R. et al, 2006):

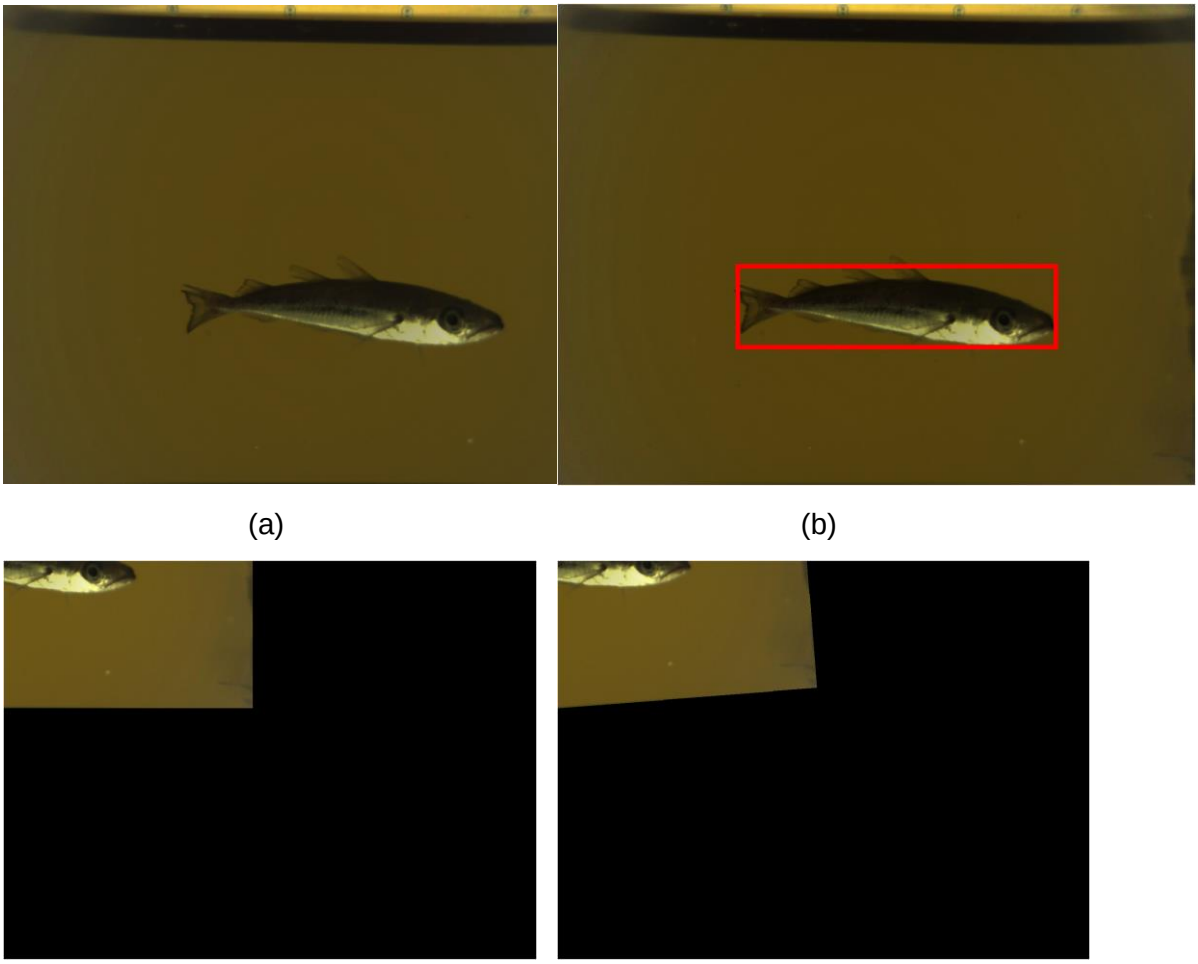
$$Homography\ matrix = \begin{vmatrix} s \cdot (Rot_{2\times2}) & (Trans_{2\times1}) \\ (0 \quad 0) & 1 \end{vmatrix}$$

on s es correspon al factor d'escala, $Rot_{2\times2}$ és una matriu de rotació 2D i $Trans_{2\times1}$ és un vector de translació, on el primer element es correspon a la translació en *x*, i el segon a la translació en *y*. Això ens permet definir homografies que transformin aquests paràmetres de forma controlada.

Amb l'objectiu de limitar les transformacions de que aplicarem a cada instància de peix, s'ha definit un domini per cada un dels tipus de transformació per separat.

En primer lloc, i per tal de facilitar la definició del domini dels paràmetres de transformació i de la implementació del codi, s'ha decidit normalitzar la posició i l'orientació de les etiquetes abans de aplicar la transformació aleatòria. Pel procediment de la normalització, en primer lloc es busca el *bounding box* o quadre delimitador de la regió, el centre d'aquest i la orientació. El *bounding box* representa el requadre de mida mínima que enquadra la regió i la orientació representa l'angle entre l'eix d'abscisses i l'eix principal de l'el·lipse que té el mateix segon moment que la regió. Aquesta informació s'utilitza per generar dos matrius de transformació, una per portar la regió al centre de coordenades i una altre per rotar-la perquè tingui una orientació de 0º respecte l'eix d'abscisses, tal i com mostra la *Figura 30.*

La rotació no implica que el peix sempre miri cap en la mateixa direcció (cap a la dreta), però se sap que la majoria de peixos que apareixen en les imatges utilitzades per extreure instàncies de peixos tenen la mateixa orientació, raó per la qual es pot assumir que la gran majoria de peixos miraran cap a la dreta després de la normalització.



(a) (b)

(c) (d)

Figura 30. Normalització de les regions abans d'aplicar la transformació aleatòria sobre la imatge. (a) Imatge original. (b) Buscar la *bounding box* sobre una instància de peix i trobar el seu centre. (c) Resultat d'aplicar la translació per portar la regió a l'origen de coordenades. (d) Resultat de rotació per mantenir el segon moment de la regió a 0º sobre l'eix d'abscisses.

Partint de que s'han obtingut les dues matrius de transformació a partir de la normalització amb la que s'envia la regió a l'origen de coordenades i aplica una rotació perquè mantingui una orientació horitzontal, es pot passar a definir el domini que limita els paràmetres que han de representar la matriu de transformació aleatòria. Aquest és el plantejament que s'ha seguit per a cada una de les transformacions bàsiques:

- **Translació**: Partint de que el centre del *Bounding box* que enquadra la regió es porta a l'inici de coordenades, s'ha decidit que el domini de la translació en els eixos x i y sigui com a màxim la mida de la imatge. D'aquesta forma, tenint en compte que la translació s'aplica sobre l'origen de coordenades que en aquest cas és el centre del *bounding box*, la regió sempre tindrà el centre dins de la imatge mantenint com a mínim la meitat del quadre delimitador per cada eix.

$$\text{Domini translació } (x,y) = [(0,0) \ldots (\text{Imatge}_{Xmax}, \text{Imatge}_{Ymax})]$$

- **Canvi d'escala**: Canviar l'escala té impacte sobre la qualitat i informació de la regió. En cas de disminuir-la, alguns píxels desapareixen, perdent informació. En el cas d'augmentar l'escala, s'han d'afegir píxels amb un mètode d'interpolació, donant lloc a una qualitat inferior proporcional al nivell d'increment de l'escala. Per aquesta raó s'ha decidit utilitzar com a màxim un valor aleatori d'escalat entre el ±10% de la mida inicial de la regió. Aquest domini s'ha buscat de forma empírica comparant les imatges resultant amb les reals.

$$\text{Domini escalat} = [0.9 \ldots 1.1]$$

- **Cisallament**: Un valor alt a la hora d'aplicar un cisallament pot distorsionar molt la regió per lo que s'ha decidit utilitzar valors de cisallament petits. L'angle de cisallament màxim s'ha definit com a 5º. Aquest efecte requereix d'una homografia a la que la part de rotació i escalat no és una matriu ortonormal amb determinant 1, tal i com es presenta més avall en aquest document.

$$\text{Domini cisallament} = [0 \ldots\ldots 5]$$

- **Rotació**: Les imatges reals demostren que la majoria dels peixos neden en la mateixa direcció, que en aquest cas és cap a la dreta. Tenint en compte que després de la normalització es parteix d'una regió amb la mateixa orientació que l'eix d'abscisses, s'ha utilitzat una distribució exponencial per generar valor aleatoris entre [0 .. 0,5] i una tria de signe per calcular una magnitud de rotació, com es mostra a la *Figura 31*. D'aquesta forma es manté la orientació cap a la dreta en la gran majoria de casos. La fórmula utilitzada per calcular els graus de rotació de la regió és la següent:

$$Graus\ de\ rotació = \ rexp * 360 * sign$$

On $rexp$ representa el número obtingut aleatòriament sobre una distribució exponencial compresa entre [0 .. 0.5] amb major probabilitat a mesura que s'apropa al zero. Aquest valor multiplicat per els 360 graus resultarà en un valor entre [0 .. 180] que es multiplicarà per $sign$, una variable que pot agafar els valors [-1,1] i que indicarà el sentit en el que s'aplica la rotació. Amb la utilització de la variable de sentit, el domini de la rotació passa a ser de [-180 .. 180] graus que inclou totes les orientacions.

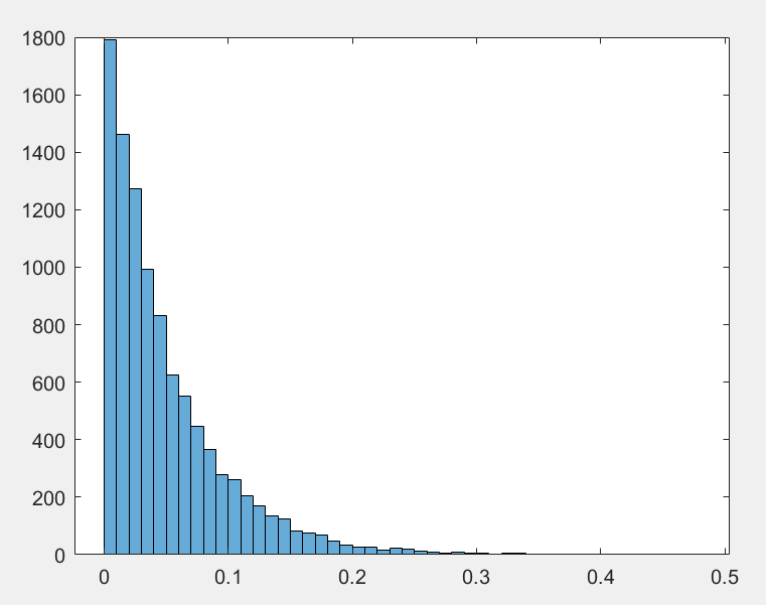


Figura 31. Càlcul aleatori del valor de rotació. Gràfica que representa 10.000 valors aleatoris que ha retornat la funció utilitzada per generar la variable $rexp$. Es pot observar com segueix una distribució exponencial i que la majoria de valors cauen entre [0 .. 0.1] que equivaldria a [0 .. 18] graus.

Un cop s'han generat els valors aleatoris de les transformacions per separat, s'ha de crear la matriu de transformació homogènia que les inclogui. La matriu resultant representa una projecció afí i té aquesta forma:

$$A_{Affine\ projection} = \begin{pmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{pmatrix}$$

On $\begin{pmatrix} c \\ f \end{pmatrix}$ representa la translació en els eixos $\begin{pmatrix} x \\ y \end{pmatrix}$ respectivament. La últim fila $(0 \quad 0 \quad 1)$ és la que s'afegeix per representar la matriu en coordenades homogènies.

Finalment la matriu $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ inclou la resta de transformacions. Per fusionar les transformacions d'escalat i de rotació s'ha fet ús del *Singular Value Decomposition (SVD)* o descomposició en valors singulars. La *SVD* és la factorització d'una matriu en la forma:

$$M = UDV^T$$

Suposant que M és una matriu de dimensions $m \times n$ i que una matriu és unitària si la seva matriu transposada conjugada és també la seva matriu inversa,

- U és una matriu unitària amb dimensions $m \times m$.
- D és una matriu diagonal de dimensions $m \times n$
- V és una matriu unitària $n \times n$

A aquesta factorització es pot afegir V^tV que és equivalent a la matriu identitat,

$$M = UV^T\ V\ DV^t$$

L'assignació de les matrius que representen les transformacions al resultat del *SVD* s'ha fet de la següent manera:

- **Canvi d'escala**. On S_x representa el canvi d'escala en l'eix d'abscisses i S_y en l'eix d'ordenades.

$$D = \begin{pmatrix} S_x & 0 \\ 0 & S_y \end{pmatrix}$$

- **Rotació**. On σ representa l'angle de rotació.

$$UV^t = \begin{pmatrix} \cos(\sigma) & -\sin(\sigma) \\ \sin(\sigma) & \cos(\sigma) \end{pmatrix}$$

- No s'assigna cap matriu de transformació a la matriu V^t raó per la qual se li assigna la matriu identitat,

$$V^t = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

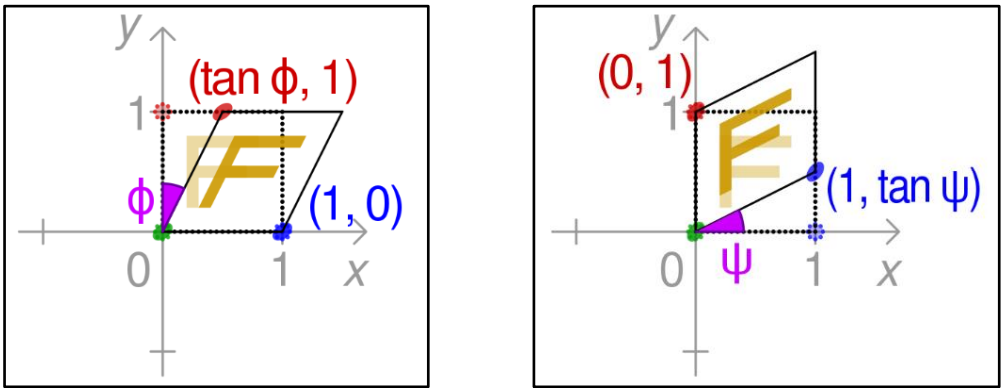
La nova matriu M es substitueix a la matriu A juntament amb els valors de la translació, donant lloc a una sola matriu homogènia que inclou totes les transformacions aleatòries menys el cisallament.

$$A = \begin{pmatrix} & M & & Tx \\ & & & Ty \\ 0 & 0 & 1 \end{pmatrix}$$

Cisallament. La transformació que representa el cisallament no l'assignem a cap de les matrius resultant de la factorització, sinó que es multiplica com una matriu de transformació sobre la que s'ha generat a partir de les altres transformacions(A). La matriu homogènia que representa la transformació de cisallament és la següent:

$$ShearTran = \begin{pmatrix} 1 & \tan(\varphi_x) & 0 \\ \tan(\varphi_y) & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

On φ_x i φ_y representen els angles de cisallament en les direccions x i y respectivament. Una representació gràfica dels angles es mostren a la *Figura 32*.



(a) (b)

Figura 32. Representació gràfica dels angles de cisallament. (a) Representa el cisallament generat donant un angle $\varphi_x > 0$. (b) Representa el cisallament generat donant un angle $\varphi_y > 0$.

Finalment, al tenir totes les transformacions en matrius homogènies, es multipliquen les dues matrius generades amb el pas de normalització amb la matriu A i la matriu de que representa el cisallament . Per tal de mantenir l'ordre d'aplicació de les transformacions la multiplicació, es procedeix de la següent forma:

$$transformation\ matrix = A * ShearTran * \ ORot \ * OTran$$

On $OTran$ i $ORot$ representen les matrius de translació i rotació obtingudes en el pas de normalització.

7.4 Inserció

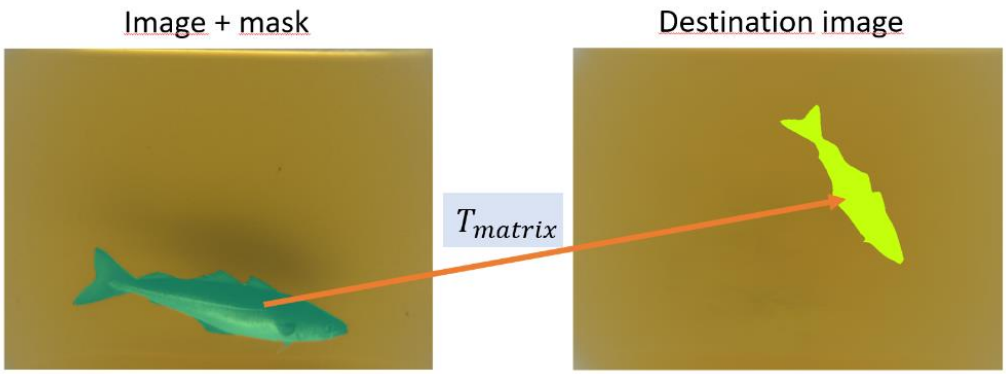
Per inserir una regió en una imatge nova, es necessiten la imatge i màscara que contenen el peix etiquetat i una imatge de destinació. La imatge de destí no necessàriament ha de ser buida, sinó que pot incloure peixos segmentats, i és per aquesta raó que s'utilitza també la corresponent màscara.

Disposant de les imatges amb les màscares i la matriu de transformació hi ha formes diferents per inserir la regió en la nova imatge. Els casos que s'han estudiat en aquest projecte són els següents:

- **Transformació directa**: Nom que s'ha donat a la transformació en la que s'aplica la matriu de transformació directament sobre els píxels de la imatge que pertanyen a la regió canviant en la imatge de destí els píxels corresponents. En aquest tipus de transformació s'han de tenir en compte dos factors importants:

    - Pot ser que el destí d'un píxel al aplicar la transformació no caigui en un píxel concret sinó entre dues de diferents. En aquests casos cal importar el valor als dos píxels o triar de forma automàtica a quin píxel s'assigna.

    - Regió resultant de la transformació amb píxels dispersos. Un altre problema que s'ha trobat amb aquest tipus de transformació és que els destins dels píxels no sempre representen el mateix bloc compacte com la regió inicial deixant píxels sense canviar en la imatge destí, com es pot observar a la *Figura 33.*



(a) (b)

$$\begin{pmatrix} X' \\ Y' \\ 1 \end{pmatrix}^{D} = {}^{D}A_{O} * \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix}^{O}$$

(c)

Figura 33. Transformació directe. Per cada píxel de la màscara representat en verd en la imatge original (a), s'aplica la matriu de transformació (c) i es canvien els píxels obtinguts per la transformació en la imatge de destí (b) per els valors RGB de la imatge original (a). La D i la O en la matriu de transformació representen el sistema de coordenades destí i origen respectivament.

- **Transformació inversa**: Nom que s'ha donat al procés en el que primer s'aplica la transformació a la màscara, s'apliquen operacions morfològiques a la regió resultant per evitar els píxels dispersos descrits en la transformació anterior i finalment, per cada píxel de la regió nova s'aplica una transformació invertida que indica quin píxel ha de recuperar de la primer imatge, com es mostra a la *Figura 34*.

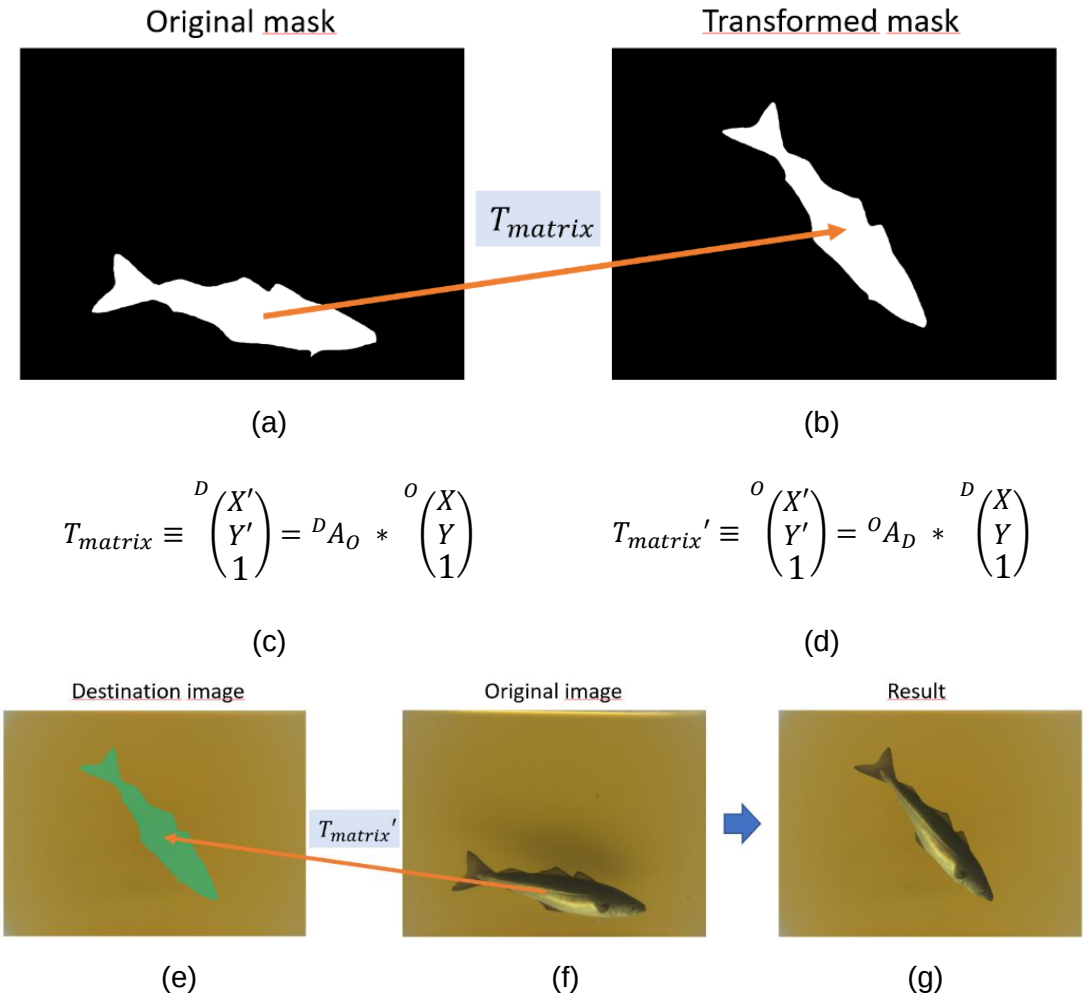


Figura 34. Transformació invertida. El primer pas és aplicar la matriu de transformació (c) sobre la màscara de la instància de peix original (a) generant una nova màscara a la que s'aplica operacions morfològiques per arrodonir i emplenar la regió (b). Per cada píxel d'aquesta màscara es canviarà el valor RGB de la imatge de destinació (e) per el valor que s'obté buscant el píxel corresponent a la imatge original (f) utilitzant la inversa de la matriu de transformació (d). La imatge resultant és la mostrada en (e).

S'ha triat utilitzar la transformació inversa al necessitar només un processament sobre la regió transformada sobre la màscara binària, per solucionar possibles artefactes que puguin sorgir, com podria ser l'aparició de píxels no assignats dintre de la regió, a diferència de la transformació directe que necessita l'ús d'un mètode d'interpolació al treballar sobre una imatge RGB. Això és possible ja que per definició, totes les homografies són transformacions invertibles.

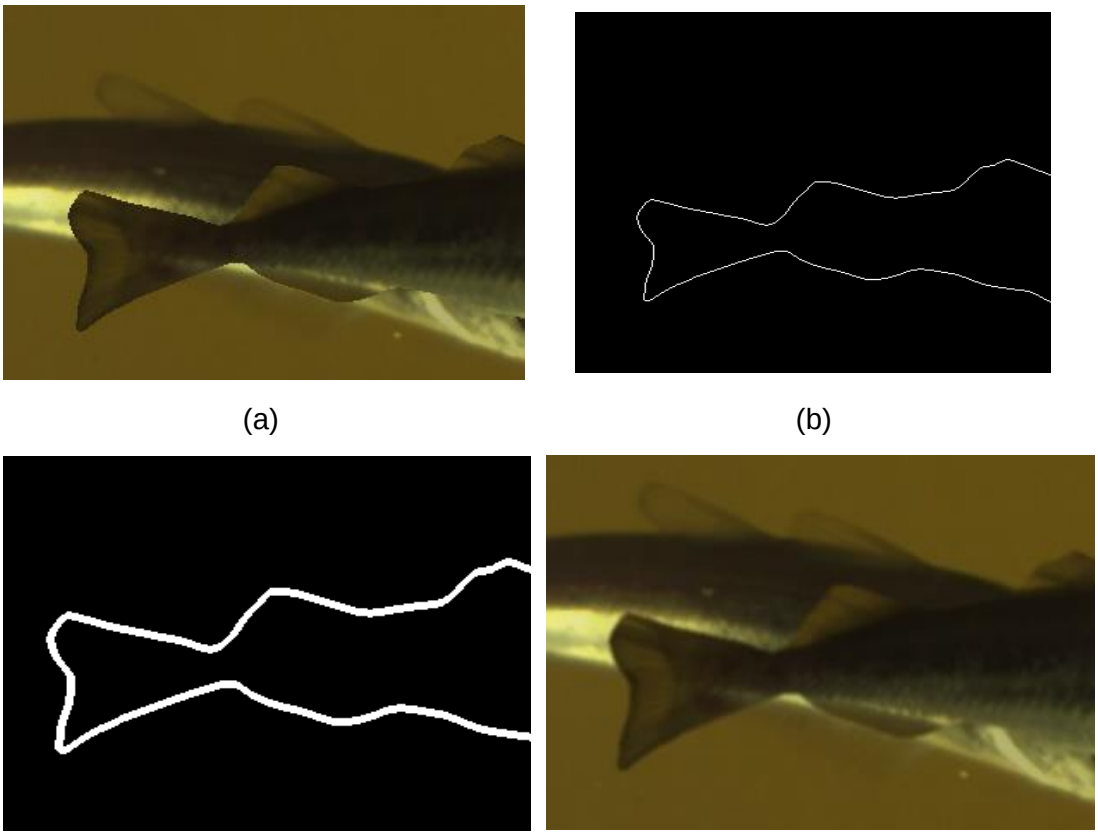
7.5 Postprocessament

Les instàncies de peix que s'insereixen en les noves imatges, després d'aplicar els processos de transformació descrits anteriorment, presenten un contorn molt definit, tal i com es pot veure a la *Figura 35*.

Figura 35. Contorn marcat resultant de la inserció d'una regió.

Per solucionar aquest problema s'identifiquen els píxels que pertanyen al perímetre afegint, així com també els que queden a cert distància d'aquests, aplicant un o diversos cops l'operació morfològica *dilate* o dilatació. Sobre aquests píxels s'aplica un filtre lineal de dues dimensions per suavitzar el contorn, com es pot veure a la *Figura 36*. S'ha utilitzat un filtre gaussià implementat en MATLAB®, trobat de forma empírica i representada per la següent matriu:

$$Low\ pass\ gaussian\ filter = \begin{pmatrix} 0 & 0 & 0.0002 & 0 & 0 \\ 0 & 0.0113 & 0.0837 & 0.0113 & 0 \\ 0.0002 & 0.0837 & 0.6817 & 0.0837 & 0.0002 \\ 0 & 0.0113 & 0.0837 & 0.0113 & 0 \\ 0 & 0 & 0.0002 & 0 & 0 \end{pmatrix}$$

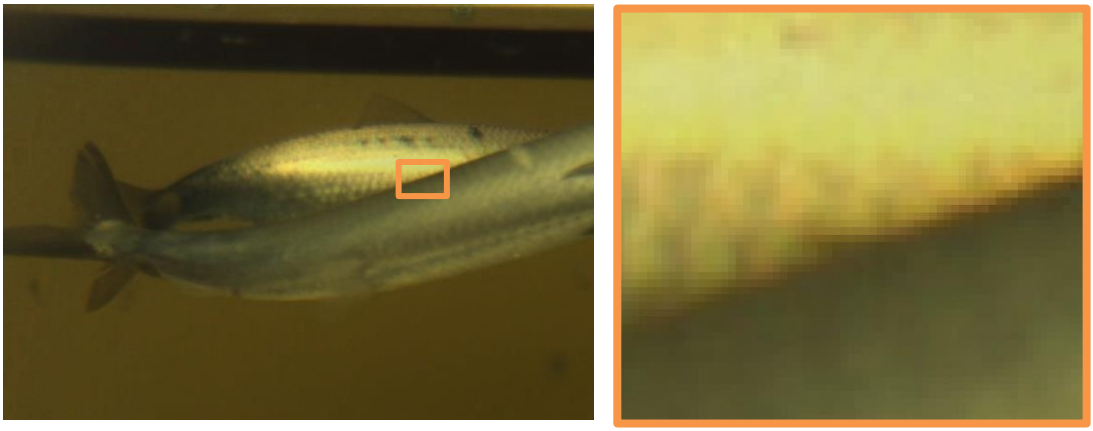


(a) (b)

(c) (d)

Figura 36. Procés de difuminat del contorn d'una regió importada. (a) Imatge que conté un peix importat sense postprocessament. (b) Identificació del perímetre de la regió. (c) Resultat després d'aplicar un *dilate* sobre el perímetre trobat. (d) Resultat després d'aplicar un filtre

En una comparació visual qualitativa entre els contorns dels peixos en imatges reals i les generades sintèticament amb aquest procediment es pot assumir que són semblants, com es pot veure a la *Figura 37*.



(a)

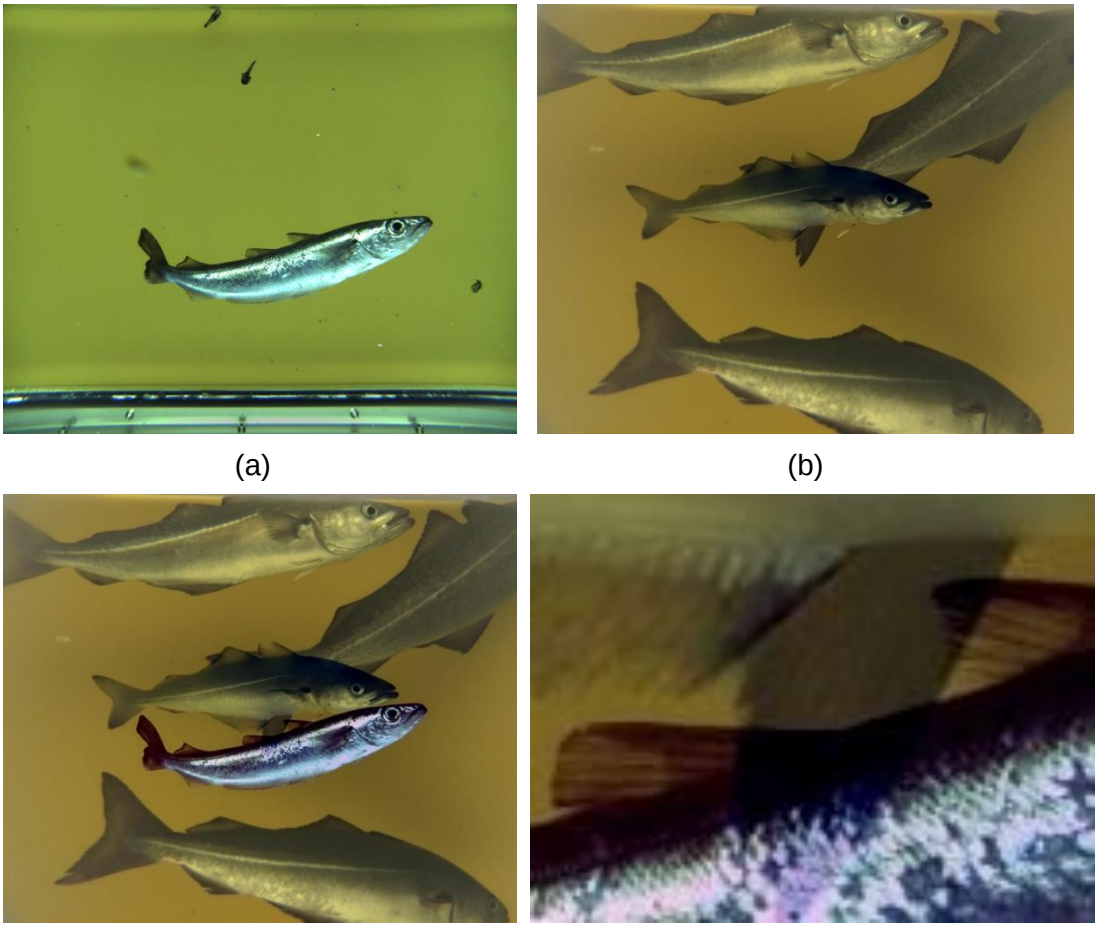
(b)

Figura 37. Comparació dels contorns de peixos entre una imatge real (a) i una imatge sintètica (b).

7.6 Estudi i proves amb mètodes de *blending*

Abans de donar com a definitiva la decisió de l'ús del mètode proposat en l'apartat anterior per eliminar el contorn marcat que queda al inserir una instància de peix en una nova imatge, s'han estudiat i fet proves amb mètodes de *fusionat d'imatges* o *blending*. Aquest mètodes són:

- **Mixed Seamless Cloning** proposat per (Pérez *et al.*, 2003). En aquest article es proposa un mètode que aplica interpolacions basant-se en la resolució de equacions de *Poisson* amb el que es creen algunes eines de processament d'imatges, i un d'ells és el M*ixed Seamless Cloning*. Els resultats, que es mostren a la *Figura 38*, s'han generat amb imatges de dos datasets diferents amb una qualitat alta tenint en compte de que tenen tonalitats i intensitats diferents, però dona lloc a un efecte de transparència que no serveix per la finalitat desitjada.



(a) (b)

(c) (d)

Figura 38. *Mixed seamless cloning*. (a) Imatge que conté el peix a transferir. (b) Imatge destí. (c) Resultat obtingut amb el mètode. (d) Ampliació d'una part que presenta transparència.

- **Modified Poisson Blending** proposat per (Afifi et al, 2015), que aplica un *Poisson Blending* semblant al utilitzat en l'apartat anterior però tenint en compte la dependència dels píxels dels contorns de la imatge transferida i de la imatge a la que es transfereix, a diferència al clàssic que només busca la dependència en els píxels del contorn de la imatge de destí. Els resultats obtinguts presenten una petita aura, quasi imperceptible, com es mostra a la *Figura 39,* però que suposaria un biaix en les dades d'entrenament de la xarxa.

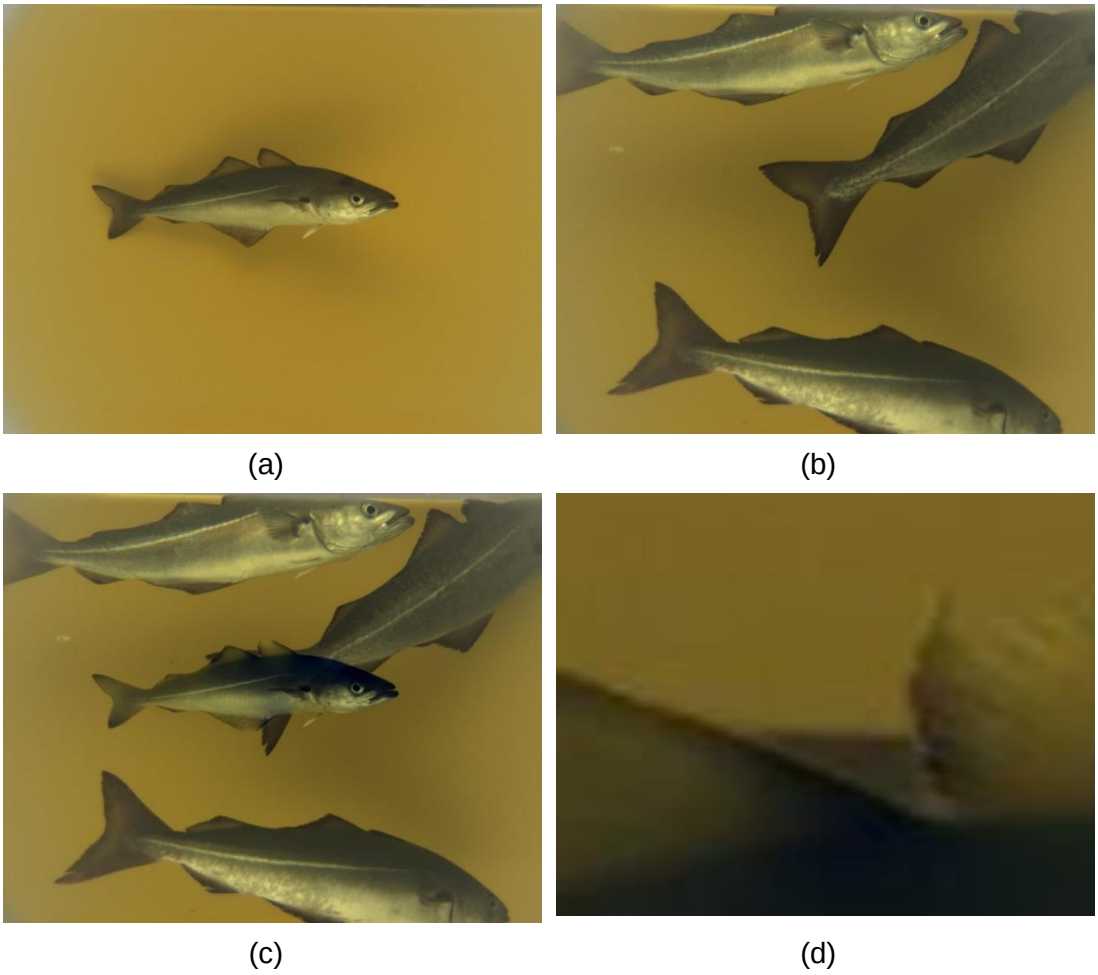


Figura 39. *Modified Poisson Blending*. (a) Imatge que conté el peix a transferir. (b) Imatge destí. (c) Resultat obtingut amb el mètode. (d) Ampliació de la imatge resultant per fer visible l'aura que envolta l'aleta del peix i que introduiria biaix a l'hora d'entrenar.

S'han provat algunes opcions més, com és el cas del algorisme *Multiresolution Laplacian Pyramid,* que es pot trobar implementat en codi MATLAB®, però en tots els casos, incloent els dos explicats prèviament, les imatges resultants necessitaven algun postprocessament. La decisió d'utilitzar transformacions amb matrius homogènies seguit del suavitzat del contorn es va fer per la facilitat d'implementació, el poc cost computacional que requereix i per la qualitat dels resultats comparant amb les imatges reals.

8. Re-entrenament del Mask R-CNN

A l'hora d'entrenar xarxes neuronals el hardware té un impacte important sobre el temps d'execució, així com també el té la grandària màxima de la xarxa entrenada. Respecte al temps, un exemple clar és la diferència que representa entrenar sobre un CPU o GPU, que implica executar les operacions de mode seqüencial o en paral·lel respectivament, i que al treballar amb convolucions, que són fàcilment paral·lelitzables representa una diferència molt significativa pel que fa al cost computacional.

El codi que s'ha utilitzat per treballar amb l'arquitectura Mask R-CNN és l'implementat per (Abdulla W., 2017) i que es troba disponible a Github. Aquest codi s'ha implementat sobre Python 3, Keras i Tensorflow.

L'entorn sobre el que s'ha treballat durant el projecte és un ordinador que consta de un targeta gràfica Nvidia RTX 2060 de 6GB, un processador AMD Ryzen 5 3600X i un sistema operatiu Windows 10. L'execució del codi del Mask R-CNN s'ha fet sobre un Pyhton 3 natiu. Aquest entorn ha suposat una limitació en dos camps:

- **Preprocessat**: Aquesta implementació del Mask R-CNN està dissenyada per executar paral·lelament, sobre els diferents nuclis del processador, una sèrie d'operacions de preprocessat. Aquest procediment és funcional a Linux, però no s'ha aconseguit fer funcionar sobre Windows, requerint l'adaptació un temps de recerca i modificacions del codi per solucionar errors generats i alentint el procés d'entrenament en utilitzar només un nucli.

- **Memòria de la GPU**: La memòria ha suposat una limitació sobre la quantitat de nivells de la xarxa neuronal que es poden entrenar simultàniament. Com més gran és la xarxa, més paràmetres conté i més espai ocupa. L'entrenament s'ha efectuat sobre una arquitectura Resnet50 com a base i modificant només les capes a partir del bloc 3 per poder carregar tots els pesos en la memòria disponible de 6GB. En intentar entrenar amb més capes, el programa funciona fins a saturar la GPU, fent saltar un error i acabant l'execució.

Per tal de re-entrenar la xarxa es necessita alimentar l'algorisme del Mask R-CNN amb un dataset que contingui:

- **Dades d'entrenament**. Imatges i corresponents màscares de *groundtruth* que la xarxa utilitzarà per treure característiques que la permetin aprendre a identificar els peixos. Com més imatges tingui aquest dataset i millor representin els diferents escenaris que es poden produir, millor serà el rendiment del model entrenat.

- **Dades de validació**. Imatges i corresponents màscares que s'utilitzaran per calcular el rendiment de la xarxa a cada iteració d'entrenament, i que permeten donar informació sobre l'entrenament que permet, per exemple, indicar que s'està fent un *overfitting* (la xarxa està començant a fallar a l'hora de generalitzar la informació apresa). Aquestes imatges han de ser reals (no sintètiques), per donar una representació correcte del rendiment de la xarxa.

Per poder comparar correctament la millora en el rendiment que representa la utilització d'imatges sintètiques, s'ha definit un dataset de prova que inclou imatges amb peixos aïllats i superposats sobre el que s'aplicarà la inferència dels models entrenats i es calcularan les mètriques. Els resultats obtinguts sobre un mateix dataset permeten quantificar la millora que representa un model respecte l'altre.

Per l'entrenament i la validació dels models que permetessin calcular la millora del rendiment a la hora d'utilitzar imatges sintètiques, s'han recopilat dos datasets (veure *Figura 40*):

- **Dataset#1**. Conjunt d'imatges i màscares reals.
- **Dataset#2**. Conjunt que inclou imatges reals i sintètiques per l'entrenament, i que conté només imatges reals per la validació i per fer les proves. Les imatges sintètiques s'han generat utilitzant fons sense cap partícula com poden ser les escames o gambes.

Les imatges dels datasets també estan classificades en:

- Imatges que contenen només peixos aïllats.
- Imatges que contenen com a mínim un peix que presenta solapament.

DATASET#1			
	train	validation	test
Single	1240	310	172
Overlap	280	70	39

DATASET#2				
		train	validation	test
Real images	Single	1240	310	172
	Overlap	280	70	39
Synthetic images	Single	0	0	0
	Overlap	5000	0	0
TOTAL	Single	1240	310	172
	Overlap	5280	70	39

Figura 40. Datasets utilitzats per generar els resultats separats en imatges que contenen peixos aïllats (Single) o que contenen com a mínim un peix amb solapament (Overlap). (a) Dataset#1 generat només amb dades reals. (b) Dataset#2 generat amb les mateixes dades del primer dataset amb l'afegit de 5000 imatges sintètiques que presenten com a mínim un peix amb solapament.

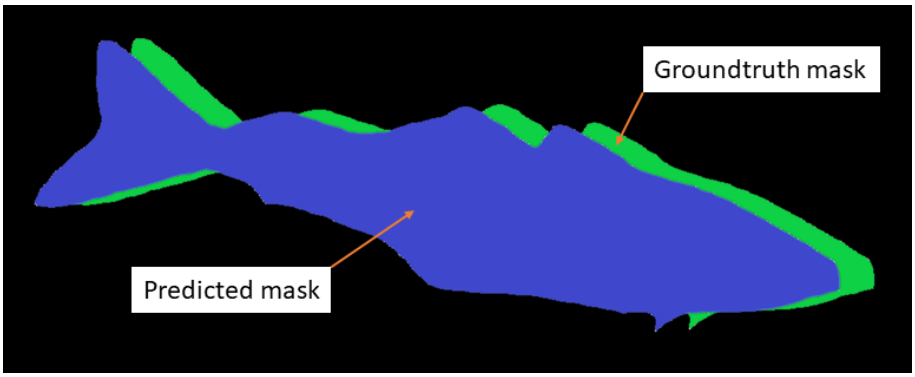
9. Mètriques

Per poder analitzar els rendiments dels models entrenats, s'han de definir en primer lloc unes mètriques amb les que es puguin mesurar diversos factors, com podrien ser l'exactitud a la hora de reconèixer peixos correctament o la precisió de les màscares generades.

Les mètriques estudiades per quantificar el rendiment dels models re-entrenades de la xarxa neuronal són MODP (*Multiple Object Detection Precision*), MODA (*Multiple Object Detection Accuracy*) i AP (*Average Precision*), explicades posteriorment de manera detallada. S'ha descartat l'ús de les mètriques MODA, MODP i AP per un error en l'algorisme implementat que descarta alguns false positives i false negatives, donant uns resultats esbiaixats.

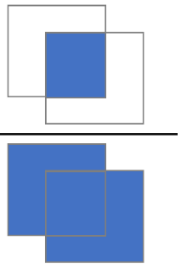
Pel càlcul d'aquestes mètriques es necessita la classificació de les prediccions (màscares generades per el Mask R-CNN) respecte el *groundtruth* en correctes o incorrectes. Al treballar amb màscares a nivell de píxel es poden utilitzar mètriques que es calculin a partir d'aquests, però en aquest cas s'ha decidit utilitzar el mètode *IoU* (*Intersection over Union*) per decidir si una predicció és correcta o no.

El mètode *IoU* (també anoment *Jaccard Index*) permet mesurar l'exactitud d'una màscara predita respecte el seu *groundtruth*, com es mostra a la *Figura 41.* El valor resultant d'aquest mètode es trobarà entre [0 .. 1], indicant una millor segmentació com més alt sigui. Un valor màxim representaria una segmentació perfecta per part de la xarxa neuronal.



(a)

$$IoU = \frac{Intersection\ area}{Union\ area}$$



(b)

Figura 41. IoU (Intersection over Union). (a) Imatge que representa en verd la màscara generada manualment que representa el peix real i en blau la màscara que ha predit la xarxa neuronal. (b) Fórmula que representa la mètrica *IoU* amb una descripció gràfica en la que es pot veure com es tracta bàsicament de la divisió entre l'àrea d'intersecció de les màscares per l'àrea de la unió.

Per classificar una predicció entre correcta i incorrecta s'ha utilitzat la mètrica *IoU* amb un llindar, que en cas de ser superat indicarà correctesa. Aquesta classificació permet definir tres combinacions, mostrats també a la *Figura 42,* que poden presentar la relació entre el *groundtruth* i les prediccions:

- **True Positives**: Predicció que s'ha fet correctament sobre un objecte definit en el *groundtruth*. Es defineix com correcte en cas de que el valor resultant del *IoU* supera el llindar que s'ha establert.
- **False Positives**: Predicció incorrecta. Es pot donar en cas de que la predicció no pertanyi a cap regió del *groundtruth* o en cas de que el valor resultant del *IoU* és menor que el llindar establert.
- **False Negatives**: Objecte definit en el *groundtruth* al que no pertany cap predicció.
- ~~**True Negatives**~~: No es contemplen en aquest cas. Representaria una predicció correcte del fons, cosa que no aporta cap informació rellevant.

	Condició positiva	Condició negativa
Predicció positiva	True positive	False positive
Predicció negativa	False negative	~~True negative~~

Figura 42. Taula de contingència d'una classificació binària entre el *groundtruth* (condició) i les prediccions (predicció). La opció 'true negative' no es contempla al referir-se a la segmentació que representa el fons i que no aporta informació prou rellevant.

Per poder aplicar la classificació s'han buscat parelles predicció-*groundtruth* en que cada una de les màscares només pot estar assignada amb una altre. La darrera raó d'aquesta assignació és la de facilitar la classificació al poder assignar directament les imatges que no tenen parelles com a fals positiu o fals negatiu, com s'explicarà posteriorment.

El procediment per aparellar les màscares és el següent:

1. Per cada imatge de la qual es tenen les prediccions i el *groundtruth* es genera una matriu d'ocurrències, com la mostrada a la *Figura 43*, a la qual s'emmagatzema informació de la quantitat de píxels que comparteixen entre màscares.

2. Es passa la matriu d'ocurrències a una llista en que cada fila conté informació del número de la predicció en la màscara, el número de la regió del *groundtruth* i l'àrea que representa la unió de les dos regions. Les ocurrències que tenen una àrea igual a zero són descartades i eliminades i la llista s'ordena de forma descendent.

3. La llista es va recorrent des de la ocurrència amb l'àrea major a menor assignant la predicció i el *groundtruth* en cas de no estar assignades encara fins que s'acaba la llista o que s'hagin assignat totes les prediccions o totes les màscares del *groundtruth*.

4. Les prediccions que no s'han assignat són considerades falsos positius i les regions no assignades del *groundtruth* són considerades falsos negatius.

	Predicció 1	Predicció 2	Predicció 3
Groundtruth 1	6020	0	0
Groundtruth 2	0	40	26191
Groundtruth 3	0	1871	0

Figura 43. Matriu d'ocurrències. Indica quants píxels pertanyen a la unió entre les prediccions i les regions del *groundtruth*.

Per classificar les parelles de predicció-*groundtruth* trobades en el procediment anterior es calcula per cada una el valor de la mètrica *IoU* i es defineix un llindar que representi amb el *IoU* el nivell mínim de intersecció per considerar correcte la parella de màscares. Les parelles que no arriben al llindar són considerades un false positive (predicció) i un false negative (*groundtruth*) i les que sí tenen un *IoU* major o igual al llindar són considerades com a true positives.

Les mètriques MODA, MODP i AP són utilitzades per puntuar el rendiment de xarxes neuronals enviades a *challenges* (concursos) en les que es busquen innovacions en l'àmbit de segmentació d'instàncies, detecció d'objectes i seguiment d'objectes com ho són el COCO (*Common Objects in COntext*) challenge i el MOT (*Multiple Object Tracking*) challenge. Les mètriques MODA i MODP s'han proposat per (Bernardin, *et al.*, 2008) per avaluar el MOT challenge.

AP (Average Precision). Mètrica molt utilitzada en challenges per determinar el rendiment en xarxes de detecció d'objectes o de segmentació d'instàncies, com és l'exemple del PASCAL VOC challenge en la que s'utilitza per fer el rànquing dels mètodes proposats (Everingham, M. Et al 2010). Aquesta mètrica serveix per quantificar l'exactitud d'un model i es calcula fent ús de les següents mètriques:

- ***Precision*** (precisió). Representa la fracció de prediccions correctes (true positives) sobre el conjunt de totes prediccions generades i ve donada per la següent fórmula:

$$Precision = \frac{true\ positives}{true\ positives + false\ positives}$$

- ***Recall*** (sensibilitat o exhaustivitat). Representa la fracció de de prediccions correctes sobre el conjunt d'objectes i ve donada per la següent fórmula:

$$Recall = \frac{true\ positives}{true\ positives + false\ negatives}$$

Els true positives, false positives i false negatives es calculen a partir del mètode explicat prèviament en el que es fa servir el *IoU* i un llindar.

Utilitzant les dos mètriques i les prediccions ordenades per confidència resultants del Mask R-CNN es genera una taula que conté:

- Correctesa de la predicció utilitzant el mètode IoU i un llindar.
- Precisió i *recall* calculats amb les prediccions anteriors dins la taula, incloent la predicció actual.

Amb aquesta informació es genera una corba que representa la precisió sobre el *recall* i que es sol suavitzar aplicant una interpolació a la precisió utilitzant la funció:

$$p_{interp}(r) = \max_{\tilde{r}:\tilde{r}\geq r} p(\tilde{r})$$

A la que la precisió a cada posició de *recall* s'interpola agafant la precisió màxima mesurada en els següents valors de *recall.*

La mètrica AP representa l'àrea de la corba de la taula *precision/recall* agafant valors mitjans de precisió en 11 nivells de *recall* separades equitativament [0,0.1,...,1] i és representada per la fórmula en cas d'aplicar la interpolació descrita prèviament:

$$AP = \frac{1}{11} \sum_{r\in\{0,0.1,\ldots,1\}} p_{interp}(r)$$

La utilització de tots els nivells de *recall* per calcular l'AP penalitza els mètodes que generin molts false negatives encara que la precisió sigui alta.

MODP (*Multiple Object Detection Precision*). Permet quantificar la precisió entre les parelles de màscares predites i *groundtruth* en un conjunt d'imatges, sense dependre de l'habilitat del model en reconèixer o no les instàncies de peix. Per poder utilitzar aquesta mètrica cal definir una distància que representi l'error de la màscara predita, i que en aquest cas s'ha utilitzat el valor del *IoU*. El càlcul d'aquesta mètrica es fa amb aquesta fórmula:

$$\mathrm{MODP} = \frac{\sum_{i,t} d_t^i}{\sum_t c_t}$$

On i representa un objecte en la imatge $\boldsymbol{t}$ i $\boldsymbol{c}$ el número de parelles (predicció-*groundtruth*) existents en aquesta. $\boldsymbol{d}$ representa el valor de la representant de l'error entre cada parella de màscares, que en aquest cas, és el *IoU*.

MODA (*Multiple Object Detection Accuracy*). Permet quantificar l'exactitud que té el model a la hora de detectar les instàncies de peix. Aquesta mètrica combina el rati d'error dels false positives i els false negatives respecte el nombre total de objectes existents en el conjunt d'imatges i és representada per la fórmula:

$$MODA = 1 - \frac{\sum_t (m_t + fp_t)}{\sum_t g_t}$$

On $\boldsymbol{m}$ és el número de falsos negatius (*misses* en anglès) i $\boldsymbol{fp}$ el nombre de false positives que es troben en una imatge t. El número total de objectes existents en la imatge es defineix com a $\boldsymbol{g}$.

Segmentation accuracy. Mètrica utilitzada en el VOC 2008 segmentation challenge i proposada en (Everingham, M. *et al.*, 2010) per avaluar la precisió de les prediccions, per cada tipus d'objecte. La fórmula és igual a la del mètode *IoU*:

$$Seg.accuracy = \frac{true\ positives}{true\ positives + false\ positives + false\ negatives}$$

Aquesta mètrica es calcula a nivell de píxel, que vol dir cada true positive representa un píxel ben segmentat per exemple. S'utilitzarà sobre cada imatge que contingui com a mínim una parella de predicció/groundtruth per mesurar la precisió total d'un model sobre un dataset. Per fer el càlcul sobre tot un dataset, s'aplicarà la mitjana sobre els resultats obtinguts per aquesta mètrica de cada una de les imatges del conjunt.

10. Resultats

S'han aplicat les mètriques sobre els models entrenats amb els datasets #1 i #2 *(Figura 40)* per poder confirmar si l'addició de les 5000 imatges sintètiques al primer dataset ha suposat una millora en el re-entrenament del Mask R-CNN.

Les imatges dels datasets estan classificades en imatges que contenen només peixos aïllats o peixos solapats per poder mesurar-les per separat. D'aquesta forma es poden analitzar dos objectius:

- La millora del rendiment de la xarxa a la hora de fer segmentacions sobre imatges amb peixos solapats, objectiu principal del projecte.

- El manteniment del rendiment sobre imatges amb peixos aïllats. En aquest cas no és interessant que millori el rendiment sobre imatges amb peixos solapats a costa del rendiment sobre imatges amb peixos aïllats.

Per altre banda, el resultat de les mètriques venen condicionades per el valor de llindar utilitzat per decidir sobre el resultat del mètode *IoU* si dos màscares es consideren una segmentació . Com és alt és aquest valor, més restrictiu es torna la mesura del rendiment. Una pràctica utilitzada per mesurar de forma robusta els mètodes de segmentació el càlcul de les mètriques sobre un interval de valors i calcular la mitjana dels resultats obtinguts. Un interval típic és IoU=[0,50:0,05:0,95], però en aquest cas s'ha decidit utilitzar només els valors [0,5, 0,75]. Els resultats obtinguts de les mètriques amb aquest interval es mostren en les *Taules 1* i *2*.

	IoU threshold 50			
	Dataset#1		Dataset#2	
	Single	Overlap	Single	Overlap
Seg. Accuracy	91,78	84,59	91,89	83,11
TP instances	224	138	225	145
FN instances	2	23	1	16
FP instances	3	12	4	11
Predictions	224	144	225	147
Groundtruth	226	161	226	161
nº fish IoU>=0.5	224	138	225	145
nº fish IoU<0.5	0	6	0	2

Taula 1. Resultats obtinguts amb un llindar de $IoU = 0{,}5$.

	IoU threshold 75			
	Dataset#1		Dataset#2	
	Single	Overlap	Single	Overlap
Seg. Accuracy	91,78	84,59	91,89	83,11
TP instances	220	115	220	127
FN instances	6	46	6	34
FP instances	7	35	9	29
Predictions	224	144	225	147
Groundtruth	226	161	226	161
nº fish IoU>=0.75	220	115	220	127
nº fish IoU<0.75	4	29	5	20

Taula 2. Resultats obtinguts amb un llindar de $IoU = 0{,}75$.

9.1 Peixos amb superposició

En aquest apartat s'analitzen els resultats entre el model re-entrenat amb dades sintètiques ("data augmentation") i el model re-entrenat només amb dades reals, que s'han obtingut de les segmentacions sobre imatges que contenen peixos que presenten oclusions.

La millora de la precisió de les màscares a nivell de píxel es pot quantificar amb la mètrica *Segmentation Accuracy.* Aquesta mètrica no depèn del llindar proposat per classificar les màscares amb el *IoU*, raó per la que els resultats són iguals en les dos taules. Els resultats indiquen un empitjorament petit en el rendiment del model al utilitzar les imatges sintètiques.

Per quantificar la millora de l'exactitud de les màscares generades ens basem en el número de *false positives*, *false negatives* i *true positives*, dels que es poden treure dos anàlisis:

- El model re-entrenat amb el dataset#2 té major nombre de prediccions correctes i menor nombre de peixos no segmentats (*false negatives*), indicant una millora a la hora de detectar i segmentar els diferents peixos.

- El model re-entrenat amb el dataset#2 té menor nombre de *false positives*, indicant que la millora observada amb els *true positives* i *false negatives* no és a costa d'una sobresegmentació de la imatge, amb el qual es té major probabilitat de segmentar tots els peixos.

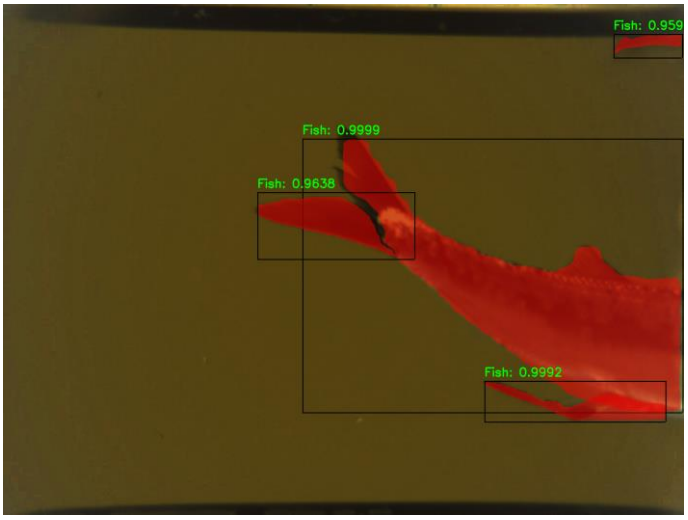
La millora definida prèviament és més visible quan s'utilitza el valor 0,75 com a llindar del *IoU.* Tenint en compte de que la precisió a nivell de píxel baixa, s'ha deduït de que possiblement alguns dels peixos que es superposen es segmentaven en el primer model com un de sol (*IoU* baix però manté la precisió a nivell de píxel) i en el segon model es segmenta cada instància de peix per separat donant un valor de *IoU* major.

9.2 Peixos aïllats

Els resultats de les mètriques sobre les imatges segmentades que contenen només peixos aïllats indiquen que el rendiment del model re-entrenat amb el dataset#2 es manté quasi idèntic respecte el model re-entrenat amb el dataset#1.

El model re-entrenat amb dades sintètiques presenta un nombre major de falsos positius. Això es deu a la alimentació del Mask R-CNN amb peixos que presenten oclusions fa que el model re-entrenat pugui arribar a interpretar peixos aïllats com dos de separats, com es mostra en la *Figura 44*.

(a)



(b)

Figura 44. Exemple de sobresegmentació. (a) Imatge original. (b) Imatge sobresegmentada.

11. Conclusions

Aquest projecte representa com la dificultat principal en l'entrenament de xarxes neuronals profundes no recau únicament en el disseny d'aquesta, sinó en la preparació d'un conjunt de dades que sigui prou gran i que representi tots els escenaris possibles per entrenar un model que generalitzi el problema a resoldre i ens proporcioni un bon rendiment.

Les mètriques obtingudes sobre els models re-entrenats demostren que la utilització de dades sintètiques representa una millora en el rendiment a l'hora de segmentar imatges que contenen peixos solapats, sense baixar el rendiment sobre la segmentació de les imatges que només contenen peixos aïllats. La millora del rendiment d'aquest model representa també l'assoliment del objectiu principal del projecte, implicant que la *pipeline* generada per passar de les imatges obtingudes del sistema Deep Vision a un model del Mask R-CNN re-entrenat amb imatges sintètiques s'ha implementat de forma correcte.

Per altre banda, hem constatat en aquest projecte que l'ús de poques imatge per fer les proves fa que els resultats no siguin prou determinants com per ser utilitzats per representar el guany en el rendiment entre els dos models re-entrenats, però ja apunten que la utilització d'imatges sintètiques senyala cap a la direcció correcta per solucionar el problema de la segmentació incorrecta en imatges de peixos sobreposats.

La feina duta a terme durant el projecte també ha servit per contribuir en l'article (Garcia, R. et al, 2019) publicat en el *ICES Journal of Marine Science*, que és una revista científica de primer nivell, indexada en el 1er quartil (Q1) al Journal Citation Reports (JCR) del Web of Science. Aquest article s'ha inclòs a l'Annex A d'aquest projecte.

12. Treball futur

Una extensió d'aquest treball podria ser la generació d'un conjunt d'imatges de test amb peixos solapats que fos més extens, permetent obtenir unes dades més consistents per fer proves i ampliar la varietat d'instàncies de peixos utilitzats en la generació de dades sintètiques.

Per altre banda, s'han recopilat algunes possibles millores aplicables sobre el mètode proposat en aquest projecte que es podrien investigar, com podria ser:

- Ús d'imatges en forma *raw* en comptes d'imatges en format *JPG* per evitar pèrdues a la hora de la compressió. La utilització de les imatges *raw* implica un augment considerable de l'espai necessari d'emmagatzematge.

- Ús d'imatges amb el color calibrat. Aquestes imatges podrien donar a la xarxa una informació més rellevant de la que donen les imatges utilitzades durant el projecte. És necessari obtenir el patró de calibratge abans de poder fer recerca sobre l'impacte que tindria aquesta solució.

- Ús de la redundància de la imatge estèreo per refinar la segmentació obtinguda per la xarxa neuronal.

- Ús del mètode TensorMask (Chen X. Et al, 2019) que genera una segmentació a nivell de píxel, a diferència del Mask R-CNN que genera la segmentació a partir de la imatge reduïda obtinguda d'un *bounding box* que representa una detecció del objecte. Aquest mètode podria millorar els contorns de la segmentació.

Finalment, l'objectiu al que apunta el projecte que envolta el sistema Deep Vision de monitoritzar la fauna que entra en una xarxa, inclou una gran quantitat de millores que s'han d'anar resolent a passos. En aquest TFG s'ha centrat exclusivament en la segmentació dels peixos, que pot ser un punt de partida per a la recerca de mètodes que permetin identificar l'espècimen i l'espècie dels peixos, mesurar-los o fer un seguiment d'aquests en imatges consecutives.

13. Bibliografia

Abdulla W. 2017. Mask R-CNN for object detection and instance segmentation on Keras and TensorFlow. github.com/matterport/Mask_RCNN.

Afifi, M., Hussain, K.F. 2015. Mpb: a modified poisson blending technique. Comput. Vis. Media.

Allken, V., Olav, N., Rosen, S., Schreyeck, T., Mahiout, T., and Malde, K. 2019. Fish species identification using a convolutional neural network trained on synthetic data. ICES Journal of Marine Science, 76: 342–349.

Bernardin, K. Stiefelhagen, R. 2008. Evaluating multiple object tracking performance: The CLEAR MOT metrics. EURASIP Journal on Image and Video Processing. 10.1155/2008/246309.

Chen X., He K., Girshick, R., Dollár, P. 2019. TensorMask: A Foundation for Dense Object Segmentation.

Everingham, M., Van Gool, L., Williams, C., Winn, J,. Zisserman, A. 2010. "The PASCAL Visual Object Classes (VOC) Challenge". International Journal of Computer Vision. 88 (2): 303–338.

Garcia-Garcia, A. Orts-Escolano, S. Oprea, S. Villena-Martinez, V. Garcia-Rodriguez, J. 2017. A review on deep learning techniques applied to semantic segmentation. arXiv:1704.06857.

Garcia, R., Prados, R., Gracias, N., Quintana, J., Tempelaar, A., Gracias, N., Rosen, S., Vågstøl, H. and Løvall, K., 2019. Automatic segmentation of fish using deep learning with application to fish size measurement, ICES Journal of Marine Science.

Girshick, R., Ren, S., He, K., Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015

Hartley, R., Zisserman, A. Multiple View Geometry in Computer Vision. Cambridge University Press, 2006.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. 2017. Mask R-CNN. In IEEE International Conference on Computer Vision (ICCV), Venice, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

Pérez P.,Gangnet M., Blake A. Poisson image editing. 2003.ACM Transactions on graphics (TOG).

Prados, R., Garcia, R., Gracias, N., Neumann, L., and Vågstøl, H. 2017. Real-time Fish Detection in Trawl Nets. In Proc. of the MTS/IEEE OCEANS 2017 Conference, Aberdeen, UK, pp. 1–5.

Salton, G., & McGill, M. J. (1986).Introduction to modern informationretrieval. New York: McGraw-Hill.

14. Annex A

S'inclou l'article publicat a l'ICES Journal of Marine Science, al qual s'ha contribuït amb el treball realitzat durant el projecte:

Garcia, R., Prados, R., Gracias, N., Quintana, J., Tempelaar, A., Gracias, N., Rosen, S., Vågstøl, H. and Løvall, K., 2019. Automatic segmentation of fish using deep learning with application to fish size measurement, ICES Journal of Marine Science.

Annex A

Automatic segmentation of fish using deep learning with application to fish size measurement

ICES Journal of Marine Science

ICES Journal of Marine Science (2019), doi:10.1093/icesjms/fsz186

Contribution to the Themed Section: '*Applications of machine learning and artificial intelligence in marine science*'

Automatic segmentation of fish using deep learning with application to fish size measurement

Rafael Garcia [1,2]*, Ricard Prados[2], Josep Quintana[3], Alexander Tempelaar[3], Nuno Gracias[1], Shale Rosen[4], Håvard Vågstøl[5], and Kristoffer Løvall[5]

[1]*Computer Vision and Robotics Institute, University of Girona, Campus Montilivi, Edif. P4, ES17003, Girona, Spain*
[2]*Girona Vision Research SL, Science and Technology Park of the University of Girona, c/ Pic de Peguera 11, Edif. Giroemprèn, ES17003, Girona, Spain*
[3]*Coronis Computing SL, Science and Technology Park of the University of Girona, c/ Pic de Peguera 11, Edif. Giroemprèn, ES17003, Girona, Spain*
[4]*Institute of Marine Research, P.O. Box 1870 Nordnes, NO-5817 Bergen, Norway*
[5]*Scantrol Deep Vision, Sandviksboder 1C, NO-5035 Bergen, Norway*

**Corresponding author: tel: + 34 676 511 024; e-mail: rafael.garcia@udg.edu.*

Garcia, R., Prados, R., Quintana, J., Tempelaar, A., Gracias, N., Rosen, S., Vågstøl, H., and Løvall, K. Automatic segmentation of fish using deep learning with application to fish size measurement. – ICES Journal of Marine Science, doi:10.1093/icesjms/fsz186.

Received 10 May 2019; revised 11 July 2019; accepted 14 August 2019.

One of the leading causes of overfishing is the catch of unwanted fish and marine life in commercial fishing gears. Echosounders are nowadays routinely used to detect fish schools and make qualitative estimates of the amount of fish and species present. However, the problem of estimating sizes using acoustic systems is still largely unsolved, with only a few attempts at real-time operation and only at demonstration level. This paper proposes a novel image-based method for individual fish detection, targeted at drastically reducing catches of undersized fish in commercial trawling. The proposal is based on the processing of stereo images acquired by the Deep Vision imaging system, directly placed in the trawl. The images are pre-processed to correct for nonlinearities of the camera response. Then, a Mask R-CNN architecture is used to localize and segment each individual fish in the images. This segmentation is subsequently refined using local gradients to obtain an accurate estimate of the boundary of every fish. Testing was conducted with two representative datasets, containing in excess of 2600 manually annotated individual fish, and acquired using distinct artificial illumination setups. A distinctive advantage of this proposal is the ability to successfully deal with cluttered images containing overlapping fish.

Keywords: deep learning, fish sizing, trawl camera system

Introduction

According to the UN Food and Agriculture Organization, 33% of commercially important marine fish stocks worldwide are overfished (FAO, 2018). One of the causes of overfishing is that, in addition to targeted species, the fishing gears often catch other unwanted fish and marine life. Globally, nearly 11% of total catches are discarded because they are not the proper species or sizes (Pérez Roda *et al.*, 2019). In some cases, the quantity of this by-catch can exceed that of the targeted species. Excessive by-catch is an immediate problem for fishers as it slows their catch sorting operations considerably, increases fuel consumption and wear on their fishing gear. Under management systems utilizing by-catch caps or closures to protect juveniles, fishing opportunities may be curtailed. In the long term, high levels of by-catch can contribute to overfishing jeopardize the long-term sustainability of the fishery.

Some countries and regions have enacted prohibitions on discarding unwanted catches. The most recent revision to the EU

Common Fisheries Policy (EU regulation 1380/2013) institutes a landing obligation requiring all catches of regulated commercial species to be landed and counted against quota. This includes catches of undersized individuals, which can be utilized to avoid waste, but not for direct human consumption or at a profit which could result in the establishment of markets.

Most fishermen use echosounders to detect fish schools and make qualitative estimates of the amount of fish and species present. Advanced "split beam" echosounders can give an indication of fish size, and characteristics such as frequency-response and school geometry can be used to differentiate between some species (Korneliussen *et al.*, 2009). However, systems to provide quantitative real-time species identification and measurement during fishing are largely in the demonstration phase (Pobitzer *et al.*, 2015; Berges *et al.*, 2018). As a result of this uncertainty, vessels relying on acoustics to target-specific species may catch undersized individuals or other species.

This paper proposes a novel fish sizing method when capturing fish using a trawl. The proposal is based on the use of the existing Deep Vision system (Rosen and Holst, 2013), directly placed in the trawl, to acquire stereo image pairs at a fixed frequency of five or ten images per second. The images are saved in a solid-state unit capable of storing ∼1 million image pairs, equivalent to 60 h of data collection. In this paper, the images have been processed offline, but we aim at processing them onboard the Deep Vision system in the near future which will make real-time active sorting possible. This will enable more sustainable fishing activities by reducing catches of undersized individuals and unwanted species.

Material and methods

Data acquisition

Data were obtained on two testing cruises in the North Atlantic, the first in the North Sea onboard the Norwegian R/V "Dr Fridtjof Nansen" during March of 2017 (hereafter dataset 1), and the second in the Norwegian Sea with the chartered fishing vessel M/S "Vendla" during May of 2017 (hereafter dataset 2). Both vessels used an 832-m circumference pelagic trawl designed for surveys of small pelagic species in the Northeast Atlantic. Dataset 1 included images of saithe (*Pollachius virens*), blue whiting (*Micromesistius poutassou*), redfish (*Sebastes* spp.), Atlantic mackerel (*Scomber scombrus*), velvet belly lanternshark (*Etmopterus spinax*), and Norway pout (*Trisopterus esmarkii*), while dataset 2 included images of Atlantic mackerel, blue whiting, and Atlantic herring (*Clupea harengus*).

Acquisition of stereo image pairs of fish in the trawl was done using the Deep Vision system which is currently used to provide fisheries survey operations with information about depth and position of fish entering the sampling trawl. Using Deep Vision, it is also possible to conduct surveys which retain images rather than the actual fish. This lessens the environmental impact of the sampling and the workload of handling and measuring the catch. At the same time it provides images and metadata that can be used for length measurements and species classification. Combined with acoustic measurements this information provides higher confidence data used as input for stock assessment.

The Deep Vision system is divided into a subsea system and a topside system. The subsea system has a stereo camera, strobe lights, battery, and an enclosing studio frame designed for optimal image quality and consistency. The studio frame is integrated into the trawl to ensure smooth flow of catch through the system, and protects the electronic components from the rigours of trawl handling and operations (see Figure 1).

The topside system provides a graphical user interface for size measurement and species classification, through a combination of manual and more automated processes. The output from the analysis software is combined with the data from the subsea system into an annotated dataset that can be used to produce statistical data.

During both surveys, the stereo image pairs were recorded at 5 fps, in JPG format, with an image resolution of 1392 × 1040 pixels. Lighting was provided by two synchronized strobes producing ∼18 000 lumen each at a colour of 4100 K. Although the lights were pointed to the ceiling and floor of the studio frame to provide diffused lighting, their angle varied slightly between cruises resulting in slight differences in reflection and illuminance inside the volume where objects pass through the Deep Vision canal (Figure 4). In addition, the user was allowed to make changes to camera exposure time, gain and gamma correction, introducing an additional source of inconsistency in image appearance. The impact of this uneven appearance on further image analysis prompted a full mechanical redesign of the lights to a production model with both higher total light output and fixed angle (Figure 1).

All the acquired images are analysed using the processing pipeline illustrated in Figure 2. First, images are pre-processed to correct nonlinearities and non-uniform lighting effects. Next, we use a Mask R-CNN architecture to localize and segment every individual fish in the image. The obtained segmentation is then refined in the next step using the local gradient to estimate the boundary of every fish. Finally, the length of the fish is computed exploiting stereo information. The different processing phases are detailed below.

Image pre-processing

Image pre-processing aims at correcting non-uniform lighting to produce images with a similar contrast between the fish and the background regardless of the location of the fish in the image. To carry out this correction, we should first linearize the image (Prados *et al.*, 2017).

Linearization is a desirable pre-processing step since cameras provide RGB values that are non-proportional with the incoming light energy. This is so because the human visual system has a nonlinear response (Burton, 1973). If an image encodes light in a [0,255] interval, a value of 128 is perceived as half the lightness by the human eye, but in reality that point is reflecting (∼) 25% of the light. That is, the camera response functions for all the colour channels are adapted to the human eye, and therefore they are nonlinear, especially if images have been stored using the JPG format, as it is often the case to minimize disc space to store large datasets. Therefore, since most processing algorithms assume that the value of a pixel is proportional to the amount of light collected by that specific pixel, linearizing the image would provide a better-conditioned set of pixel values for further processing. Moreover, using linearized images ensures providing the processing algorithms with a more accurate representation of the measured spectra, and consequently its behaviour and outputs become more consistent. In our case, images are linearized using the camera linearization method described in Debevec and Malik (1997). After this process, the RGB values become proportional with the irradiance on the sensor pixels, and the image is ready to

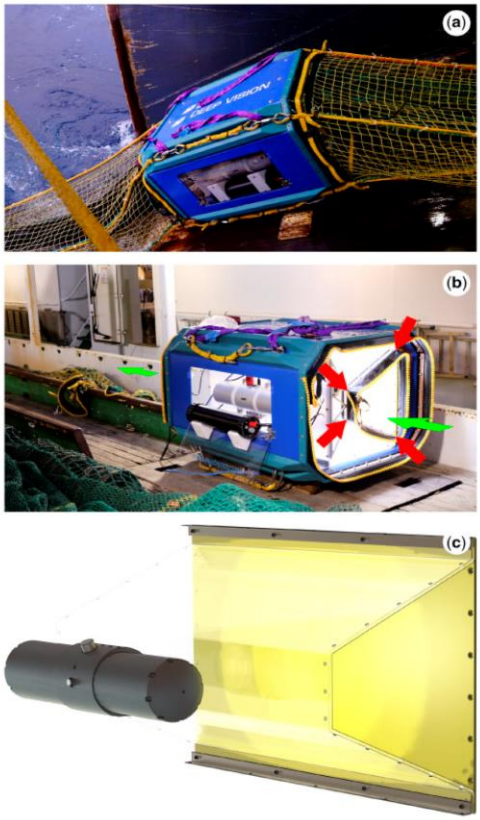


Figure 1. Deep Vision subsea system. The system is placed inside a trawl net (a) and contains a stereovision camera set and indirect lighting source. The arrows in the middle figure (b) define the "studio" section, corresponding to the area where the catch flows, and which can be seen in detail in the bottom schematic (c). Fish cross through a trapezoidal plexiglass section which ensures they maintain at least 20 cm distance from the cameras and lights and are within the field of view of the cameras.

undergo further linear operations, such as the correction of the non-uniform lightning. All subsequent operations are performed in linear RGB values.

Although the Deep Vision system provides images with a good overall illumination, the amount of light on the central area of the images is higher than that at the corners of the image. Therefore, once the images are linearized, we also correct the images for non-uniform lighting. To do this, we first convert the images from RGB to HSV (Hue, Saturation, Value), where V corresponds to the image luminance (Schwarz *et al.*, 1987). The luminance channel is the only component that will be used to correct the illumination effect. The illumination correction is performed by modelling the background, i.e. we compute the

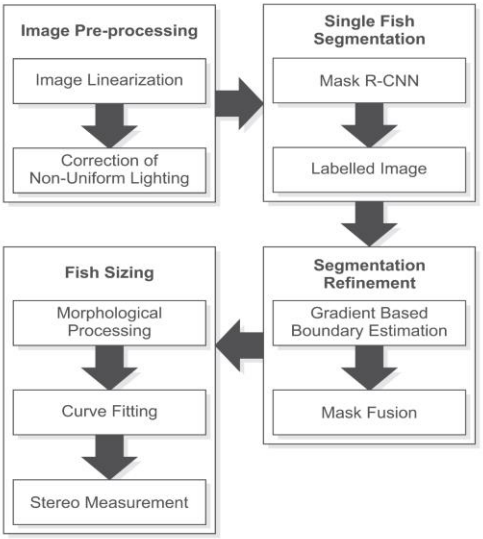


Figure 2. Automatic fish measurement pipeline. The process starts with the pre-processing of the image, and then a CNN localizes every individual fish. The CNN also provides a segmentation mask for the fish. Next, these masks are refined using local contrast information to delineate the boundary of every fish, and finally the length of the specimen is measured based on stereo cues.

median of a sufficiently large set of images of the scene (typically 300). The high power of the lighting system makes any external lighting contribution negligible, and consequently the illumination can be assumed as constant during the whole trawl. Ideally the images are selected at the beginning of the trawl haul before fish begin entering the field of view, although the only requirement is that, for the volume of 300 images, every pixel coordinate should not contain fish in slightly more than half the images ($>$150). The median value for each image pixel will be later on computed. If a given coordinate show no fish most of the time, the appropriate background value will be kept for this pixel location by the median measure. Once the median image has been computed from the V component of the set of images, we obtain a background luminance image that allows us to infer the illumination of the scene. The estimated background image is then inverted and applied as a non-uniform illumination compensation pattern to correct the luminance (V) of every image of the sequence. The RGB values of the final images are recovered from the HSV representation, ensuring that the correlation between the RGB channels is preserved, i.e. the original colours are kept.

It should be noted that working directly on the RGB colour space using channel-wise processing, as is commonly done in several image processing algorithms, may lead to a loss of the correlation between the values of the RGB triplets, thus shifting the original colours acquired by the camera.

Compensating the non-uniform illumination on all the images has proved to better condition the data to perform the subsequent fish segmentation (Prados *et al.*, 2014).

Single fish detection

Our aim is to be able to segment individual fish in the images, so that measuring the fish once it has been segmented becomes a trivial task. Figure 3 illustrates the problem we want to solve. Figure 3a shows a situation in which fish segmentation is quite easy since the background of the Deep Vision system can be modelled, and everything that is not background could be assumed to be a fish. However, Figure 3b shows a more challenging situation in which the fish to be measured are overlapping, making it difficult to determine their outline. In these situations in which we are not able to formalize an algorithm to recognize an object (e.g. a fish), using of machine learning methods has shown to be the best alternative. Among machine learning, deep convolutional neural networks (CNNs) have proved to be capable of achieving the best results on challenging datasets using supervised learning (Krizhevsky *et al.*, 2017). CNNs have also demonstrated good accuracy in automatic classification of species using simulated Deep Vision images (Allken *et al.*, 2019).

One of the state-of-the-art CNN-based deep learning object detection approaches is *Region-CNN* (or R-CNN). R-CNN provides a solution to the fast detection of regions of interest (RoI) within an image. Based on this approach, more complex architectures have recently appeared such as *Faster R-CNN* (Girshick, 2015) for faster speed object detection, as well as *Mask R-CNN* (He *et al.*, 2017) for object segmentation. In this paper, we use a *Mask R-CNN* architecture for fish detection and segmentation. *Mask R-CNN* combines *Faster R-CNN* for object detection in which the number of objects may vary from image to image, and fully convolutional networks (FCNs) for segmentation to establish what pixels in the image belong to what object. This step of detecting and delineating the boundaries of every individual object in an image is called "semantic segmentation," and allows us to differentiate individual fish when two or more instances of a fish overlap in the image, as illustrated in Figure 3b.

Faster R-CNN performs individual fish detection in two stages. First, it determines the bounding boxes (i.e. RoIs) using the region proposal network (RPN) standard. The RPN is basically a lightweight neural network that scans the image in a sliding-window fashion to find regions that contain objects. Second, for each RoI it determines the class label of the object through RoI pooling. Therefore, *Mask R-CNN* incorporates these two stages, but it performs RoI pooling in such a way that there is no loss in stride quantization due to rounding when pooling is performed, as opposed to the rounding performed by *Faster R-CNN* (Ren *et al.*, 2015). Moreover, the sliding window is handled by the convolutional nature of the RPN, which allows it to scan all regions in parallel exploiting the GPU architecture.

FCNs are used to predict the mask for every RoI. Convolutional layers retain spatial orientation and this information is crucial for location-specific tasks such as creating a mask for every individual fish (He *et al.*, 2017). This is a clear advantage with respect to fully connected layers, in which the spatial orientation of pixels with respect to each other is lost as they are squeezed together to form a feature vector (Long *et al.*, 2015).

Our Mask R-CNN architecture was initially pre-trained for the COCO dataset (Lin *et al.*, 2014). Then, the last layer was modified to classify between fish and background and we re-trained the last layers using our fish training data for 20 iterations. This fine-tuning strategy allows us to reduce the training time and the needed amount of data compared to training from scratch. Next, the full network was trained with our trawling data. In all cases, during training we tried to reduce overfitting on image data by artificially enlarging the dataset using data augmentation, which included image translations, horizontal and vertical reflections, rotations, and shear transformations.

Segmentation refinement

The mask computed by Mask R-CNN has been obtained using a low-resolution image. Thus, the mask that segments the fish has a lower accuracy than those that can be obtained from the full-resolution original images. Therefore, a final stage of mask refinement is applied to obtain a much finer spatial layout of the fish, i.e. a more accurate segmentation.

The blobs estimated by Mask R-CNN are first scaled and transferred to the full-resolution image (1228×1027 pixels). Then, the gradients of the V channel on the original image are computed. This results in an image were the boundary of the objects is clearly distinguishable. The gradient magnitudes are thresholded to keep only the higher values, that is, the most prominent boundaries. Finally, both the Mask R-CNN masks, resulting in most cases in conservative segmentation, and the gradient-based boundary refinement masks, are fused into a single one for each image object. Empty inner areas are filled using binary morphological operators.

In case of overlapping fish, Mask R-CNN masks are used to guess where the boundaries of every specimen should be placed, given that the gradient-based refinement cannot distinguish among different objects. To determine which pixel belongs to each fish, Mask R-CNN masks are dilated using a customized multi-label dilate operation, which stops growing in a given direction when another neighbouring object is growing in the opposite direction and colliding with the first. The result of this dilate operation is used to determine the contribution of the gradients image to each fish mask.

Segmentation performance

To evaluate the performance of the masks obtained by our processing pipeline, a detection accuracy measure is required. A standard set of metrics [intersection over union (IoU) and pixel accuracy] is used to quantify the segmentation results, since they are the *de facto* evaluation metrics used in object detection. IoU, also referred as Jaccard index, is an evaluation metric used to measure the accuracy of object segmentation on a particular dataset. IoU is often computed using the bounding box predicted by the CNN detector and the ground-truth (i.e. hand labelled) bounding box. In our case, since our detector generates a pixel region (mask) containing the pixels that correspond to a given fish, and the ground-truth is also a hand-labelled pixel region, IoU is computed using these two regions. The final score is obtained by dividing the area of overlap of the predicted region and the ground-truth region by the area of union of both the predicted region and the ground-truth region:

$$\text{IoU} = \frac{\text{ground-truth} \cap \text{prediction}}{\text{ground-truth} \cup \text{prediction}}.$$

However, the measure of pixel accuracy corresponds to the percentage of pixels in the image which were correctly classified.

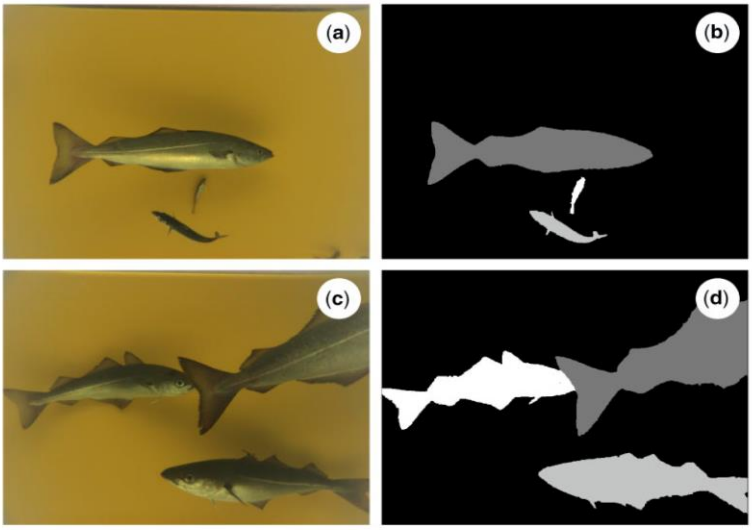


Figure 3. Fish segmentation. In simple cases such as (a), fish can be segmented into individual specimens simply by background subtraction (b). However, we need a cognitive understanding of the image to be able to segment the three fish instances in (c) shown in (d).

Usually it is presented for each class and the mean of all classes is provided. In our case both values are the same as we only have the "fish" class.

For this metric we need to introduce the notions of TP, TN, FP, and FN. True positive (TP) represents a pixel that is correctly predicted to belong to the given class whereas a true negative (TN) represents a pixel that is correctly identified as not belonging to the given class. False positives (FP) and false negatives (FN) are defined accordingly. The accuracy metric is then computed as

$$\text{accuracy} = \sum \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

Length estimation

Once the specimens have been properly segmented, the final stage consists of finding a line that accurately describes the length of the fish. For this purpose, we estimate the fish skeleton using morphological operations applied to the labelled image, but it should be noted that the actual length of the fish should be estimated taking into account its 3D pose. The thinning morphological operation involves eroding the segmented region until skeleton level (Dougherty, 1992), i.e. shrinking the region corresponding to the individual fish until the blob becomes 1 pixel wide. This typically leads to a line centred along the main axis of the fish. Before performing morphological skeletonization, the binary masks resulting from the segmentation of the previous section are smoothed by applying a "closing" morphological operation. In this way, a continuous and typically smooth line is obtained, representing the main axis of the fish.

The next step is the estimation of a curve following the trajectory described by the pixels of the skeleton. Once the points defining the skeleton have been obtained, a cubic polynomial is estimated using RANSAC (Fischler and Bolles, 1981). In this way, the points of the skeleton are classified in inliers and outliers, and after a number of iterations, a consensus solution is computed by least squares fit of the largest set of inliers, obtaining the final estimation of the curve.

Once the curve equation is derived, the starting and ending points defining the length of the fish are determined as the intersection between the estimated curve and the boundaries of the smoothed fish blob. Since the stereo system has been calibrated and the images rectified (Hartley and Zisserman, 2003), these points can then be easily transferred from the right to the left image of the stereo pair by applying the axis constraints determined by the stereo rectification. Then, once front and back points have been established in both images of the stereo pair, a set of uniformly distributed points along the curve are selected in the right image. These points are transferred to the left image following the same uniform distribution, using the image rectification to determine its Y location. Finally, the set of measurement point pairs from the right and left images is used to compute the distances of the segments connecting them using epipolar geometry, thanks to the calibration of the stereo system.

Results

A total of 1805 manually annotated images (corresponding to the left camera of the stereo pairs) have been used to validate the pipeline proposed in this paper, with a total of 2629 fish annotations. These images have been acquired in two different cruises. Dataset 1, including 1605 annotated images, was acquired by R/V Dr Fridtjof Nansen on March 2017. This dataset represents a small subset of all the images acquired during the survey, and includes frames from three different hauls (138 055 stereo pairs).

Dataset 2 was acquired by F/V Vendla on May 2017 and it includes 200 annotated images, all of them from the same haul (28 117 stereo pairs). Both surveys consist of thousands of images, but only small samples containing fish suitable for an appropriate labelling (a large percentage of images contain no fish at all) can be used. The annotation effort is significant, taking into account that the labelling procedure implies a precise manual segmentation of each specimen, not a simpler approximate bounding box specification.

Figure 4 illustrates the appearance of the images of both datasets, as well as the result of correcting non-uniform illumination. It should be noted that the appearance of the images in both datasets is different due to the change of lighting arrangement and camera parameters (with a gain factor of 1.2 in case of dataset 1 and gain factor of 2 in case of dataset 2). In dataset 2, the central part of the image is considerably brighter than in dataset 1, and as a consequence, the margins of the image are darker than in the first dataset. After applying the strategy to compensate the non-uniform lighting, using a specific per-haul pattern to maximize precision, the images of both datasets become better suited for posterior processing. The frames attain a more even appearance, with uniform light distribution, making the contained data better conditioned for the subsequent steps.

Two different sets of experiments have been conducted. In the first experiment, we aimed at evaluating the capability of the architecture to generalize the problem of fish detection by training using the 1605 images of dataset 1, and then testing on the 200 annotated images of dataset 2, in which lighting conditions and camera settings have changed.

It should be noted that the two datasets also present different characteristics in terms of the type of fish present. Saithe dominated in the first cruise, which also included blue whiting, redfish, Atlantic mackerel, velvet belly lanternshark, and Norway pout. The second cruise included images of Atlantic mackerel, blue whiting, and Atlantic herring. In addition to these fish, the second dataset also included northern krill, *Meganyctiphanes norvegica*, in most images. Moreover, the average number of fish per image is also much larger in the second dataset.

The Mask R-CNN was trained with the images of dataset 1, acquired by the R/V "Dr Fridtjof Nansen," but applying the data augmentation techniques described above. The original dataset was split into 80% for training and 20% for validating.

After finishing this training we applied the obtained weights on 200 annotated images from the second dataset acquired by F/V "Vendla." This dataset is completely independent from the images used for training and validation. Test images were previously segmented by hand, creating a ground-truth to compare all methods. Fifty of these images contain overlapping fish while the other 150 contain one or more fish, but with no overlap. Table 1 illustrates the results obtained in this first trial.

Analysing the values of Table 1, the reader would think that the CNN is doing a good job. We differentiate between "single fish," which is the detection of fish when the masks corresponding to the fish are not connected to each other (see Figure 3a), and "overlapping fish," which corresponds to the cases in which these masks overlap (see Figure 3b, central fish). In Table 1, IoU is ranging between 0.84 for "single fish" detection, and 0.82 for "overlapping fish." And the accuracy is even higher with values of >0.98 in both cases. Therefore, at first glance, the Mask R-CNN architecture seems to have done a good job to generalize the problem of fish detection.

It should be noted, however, that in our case we want to segment every isolated fish to enable its later sizing. In the case of overlapping fish (see Figure 3b), applying IoU out of the box would only take into account if a pixel that was predicted as class "fish" belongs to a fish in the ground-truth. However, this is not what we need in our application. Consider the example of Figure 5. The ideal ground-truth masks are shown in Figure 5a, with the red fish labelled as 1 and the blue fish with label 2. Figure 5c shows a fish segmentation in which the two overlapping fish are detected as a single fish. This would be considered as a very good segmentation in the standard IoU metric frequently used in the literature, e.g. (He *et al.*, 2017), but in our case we consider this a bad result since it is missing the detection of fish 2, and over-segmenting fish 1. Therefore, we introduce a new metric, namely IoU*, to measure IoU on a slightly different way that better serves our purposes. This measurement of IoU* will work as follows. An IoU* measurement will be computed for every fish in the ground-truth. The IoU* corresponding to the red fish as the area of intersection between the red region in Figure 5a and the red area in Figure 5c, and that value will be divided by the union of the same two regions. In this way, the detection of fish 1 will have a low IoU, as we will divide by a large area of union. Equally, for fish 2 we will divide the area of intersection by the total area of union of Figure 5b plus the blue area of Figure 5a, also producing a low IoU* value since it will have also a large number in the denominator. Using this metric, large values of IoU* guarantee that only one fish has been detected, while low values indicate that two or more overlapping fish in the ground-truth have been predicted as a single fish in the detection phase. Experimentally, this threshold has been set as 0.7.

The results of this new metric are given in Table 2. Again, we distinguish between the previous two cases depending on whether fish are overlapping to have a better insight of the performance of the system under this critical situation. In the first two columns the table details the number of images of the second dataset, and the total number of fish manually annotated in those images. The third column states how many of these fish are detected with an IoU* with a value of >0.7, which intuitively means that the detection is good, i.e. two fish in the ground-truth are detected as two fish in the trial, and not as a single, larger fish. For the case of single fish (non-overlapping) we observe that 334 fish are correctly detected out of the 368 fish in the ground-truth. This is really a good performance if we take into account that several of the fish manually annotated in the datasets correspond to partially visible fish that are entering or leaving the field of view of the camera. However, for the images in which fish are overlapping, only 154 out of 272 fish are detected with an IoU* >0.7. And 94 fish are detected with IoU* <0.7, i.e. one fish is detected when >1 fish appeared in the ground-truth. It can be observed that, as opposed to what it seemed in Table 1 using the standard IoU metric, the performance of Mask R-CNN in this first trial is not so great, especially in the case of overlapping fish. The next two columns present the number of false negatives, i.e. fish not detected at all, and false positive. In this dataset the false positives normally correspond to the prediction of fish in areas of the image that correspond to northern krill, present in all the images of sequence 2. Finally, the last column corresponds to the average IoU* measurement, giving a value of 0.76 for the single fish case, and 0.58 in the case of overlapping fish. It should be noted that this average is computed from all the IoU* values of all the

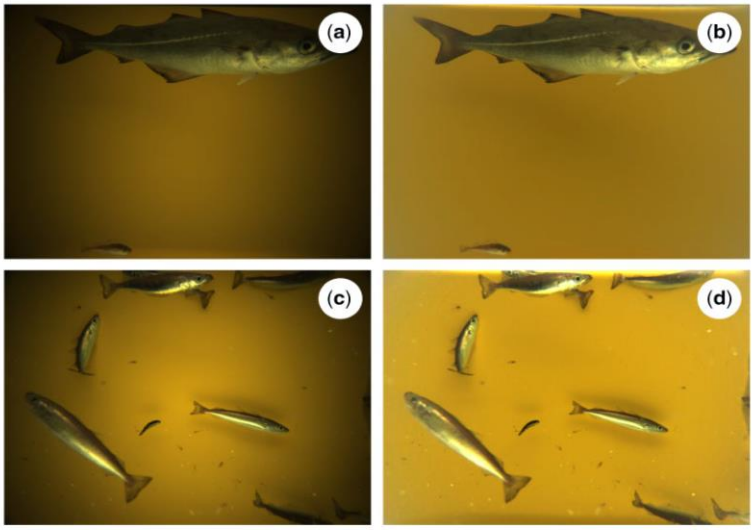


Figure 4. Correction of non-uniform illumination in dataset 1 (top) and dataset 2 (bottom). (a) Image from the Dr Fridtjof Nansen March 2017 dataset. (b) Image after non-uniform illumination compensation. (c) Image corresponding to the Vendla May 2017 dataset. Note the different appearance of the image with respect to (a). The centre of the image is brighter, while the boundary areas are still significantly dark. (d) Image after non-uniform illumination compensation.

Table 1. Results obtained by Mask R-CNN.

	IoU	Accuracy
Single fish	0.845	0.994
Overlapping fish	0.824	0.984

The network was trained using dataset 1, and the test has been quantified using the images of dataset 2. The results suggest a very good generalization capability of the network for detecting fish.

images in the corresponding dataset. We average all IoU* values for every fish in the ground-truth, but we also accumulate and account for 0 if FN or FP occur in the test images. Therefore, our average IoU* metric strongly penalizes false detections.

The last two rows of Table 2 detail the results of taking the fish detection masks obtained in this first trial by Mask R-CNN and applying the gradient mask refinement to them. We notice that gradient refinement is not able to improve fish detection, although it raises IoU* to 0.80 and 0.61, respectively. This basically means that the segmentation mask is more accurate after gradient refinement.

Table 3 reports the results of the second experiment. In this case, both datasets were used to create the train, validation, and test sets. Out of the total number of images (1805), roughly a 10% is used to evaluate the final model fit on the training dataset (test set), and the remaining 90% of the images were further divided into 80% for training and 20% for validation to tune the hyperparameters of the Mask R-CNN. Again, to better understand the performance of the network, we divided the test set images between (a) single fish and (b) overlapping fish situations. For the single fish scenario, as expected, we see that the performance of the detection is better than in the first experiment, since the training data includes images of datasets 1 and 2. More than 96% of the fish are correctly detected when there is no overlapping fish, i.e. 225 correct detections from 233 annotated fish. This percentage goes down to roughly 79% when fish are occluded by other fish. These results with overlapping fish drastically improve the results of experiment 1, with 57% of correct detections of overlapping fish. It can also be observed that the number of FN and FP has also been drastically reduced with respect to the previous trial. Finally, the last column of Table 3 includes IoU* average values of 0.89 and 0.79 for non-overlapping and overlapping fish, respectively. These values are slightly improved by the gradient refinement technique, on 0.01 in every case. This is a sign that the masks generated by Mask R-CNN in the second experiment are more accurate than the ones predicted in the first trial, but can still be improved through gradient refinement. Some sample results of the second experiment can be shown in Figures 6 and 7.

Figure 7 shows intermediate qualitative results of the proposed pipeline. It can be observed how the individual fish segmentation algorithm provides a much better fish delineation with respect to the labelled image provided by Mask R-CNN.

Discussion and conclusions

Fish length estimation and catch composition are among the most crucial information collected in fisheries research. The Deep Vision system allows fishing vessels to collect stereo imagery, and proper processing of these data enables gaining critical information about average fish size and catch composition during the trawling operation.

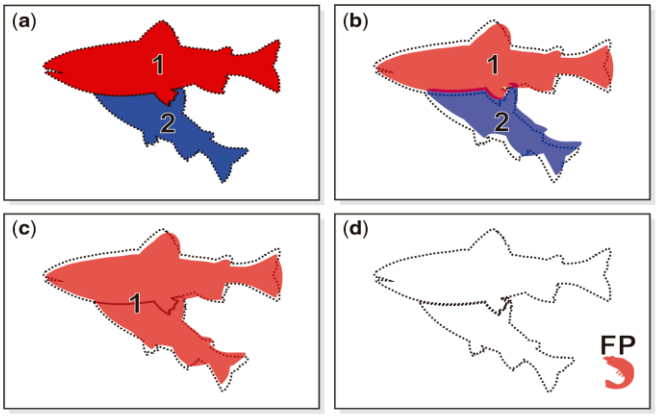


Figure 5. Fish masks. (a) Ground-truth hand annotation. (b) Example of masks detected by the CNN. The dashed lines show the corresponding ground-truth. The coloured area outside the dashed region corresponds to a false positive area, the white area inside the dashed region defines a false negative. (c) Example of an incorrect segmentation in which the CNN detects as a single instance the two fish of (a). (d) False detection of a non-existent fish, giving rise to another false positive.

Table 2. Experiment 1: results obtained by Mask R-CNN after training with dataset 1 (D#1) and testing with dataset 2 (D#2).

		No. of images	Total no. of annotated fish	No. of detected fish with IoU* >0.7	No. of detected fish with IoU* <0.7	FN	FP	IoU*
Mask R-CNN train and valid. on D#1 + test on D#2	Single fish	150	368	334	15	19	25	0.76
	Overlapping fish	50	272	154	94	24	16	0.58
Gradient refinement	Single fish	150	368	333	16	19	24	0.80
	Overlapping fish	50	272	156	95	21	15	0.61

Performance taking into account the new metric IoU* that penalizes detection of a single fish when two or more fish instances are labelled in the ground-truth.

Several works in the literature have tried to segment fish in underwater video sequences. Some achieve fish detection based on matrix decomposition (Qin *et al.*, 2014) or exploiting texture and shape features that characterize fish with respect to the background (Spampinato *et al.*, 2010). Other works rely on salient features (Fernandes *et al.*, 2016), carefully selected double thresholds (Chuang *et al.*, 2016), or the guided filter (Sanchez-Torres *et al.*, 2018). In many cases, the approach involves a static camera that allows modelling the background to then isolate the fish to carry out monocular detection or stereo measurements (Costa *et al.*, 2006; Pérez *et al.*, 2018), while other works train-specific Deep Learning architectures for fish classification (Qin *et al.*, 2016). However, in all cases the detected fish where not overlapping with other fish in the field of view of the camera. Proper delineation of individual fish in overlapping situations still remains a challenge.

Stereo imaging is often employed to obtain depth information, and depth cues can be used to segment RoI in some well-conditioned situations. However, traditional stereo matching techniques such as *Semi Global Matching* (Hirschmuller, 2005) or *Block Matching* (Konolige, 1998) fail to reliably detecting the fish boundaries in cluttered situations, as depicted in Figure 8. Depth cues from stereo alone can potentially be used to separate fish standing at clearly different distances, such as in the case of Figure 8a and b. On the contrary, we find in our datasets many cases in which multiple fish stand at approximately the same distance while overlapping, or are imaged while being significantly rotated from the ideal fronto-parallel configuration (such as in Figure 8e and f), In these situations, stereo matching fails to provide enough information to successfully and robustly separate the fish (Figure 8g and h). Figure 9 illustrates the result of our approach for this particular complicated case. While the result is not perfect in Figure 9b, it can nonetheless be considered as a successful detection and separation.

The processing pipeline proposed in this paper is able to provide accurate segmentations of individual fish in images acquired during standard fisheries surveys using the Deep Vision commercially available system. The pipeline involves three main phases: pre-processing, CNN-based segmentation, and gradient refining. Each phase contributes decisively to the performance of the overall system.

Pre-processing aims at exploiting the fact the imaging acquisition setup is well defined and constrained in terms of optical sensors, illumination characteristics, and background. By performing adequate modelling of the camera response and background illumination field, the variability of the visual appearance is reduced across different datasets and surveys. This, in turn, promotes the performance of the CNN, and, to

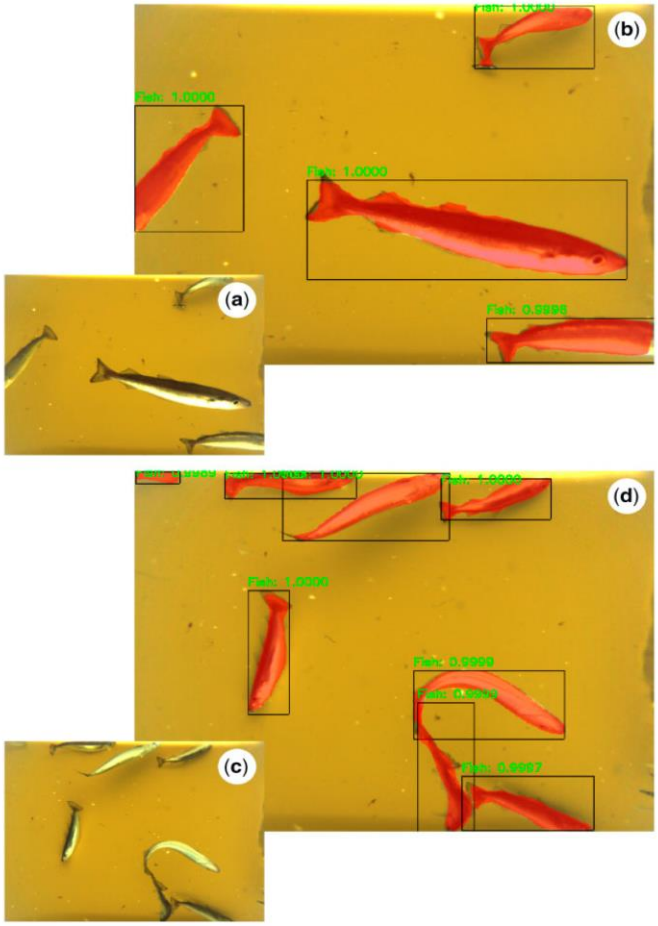


Figure 6. Fish detection and semantic segmentation performed by Mask R-CNN. (a) and (c) correspond to the original images. (b) and (d) illustrate the outcome of the algorithm. Note how Mask R-CNN is also able to detect overlapping fish, as shown in (d).

a lesser extent, also benefits the gradient refinement step at the end.

The Mask R-CNN architecture was selected for the *CNN-based segmentation*. A central reason behind this choice was its superior performance reported by He *et al.* (2017), when compared to closely related instance-aware alternatives such as Multi-task Network Cascades (Dai *et al.*, 2016) and Fully Convolutional Semantic Segmentation (Li *et al.*, 2017).

Finally, the gradient refining phase improves the delineation of the fish by using local contour cues. The impact of this step is clearly visible on Tables 2 and 3 regarding the IoU* measurement, where there was a noticeable improvement. The improved delineation is also of clear benefit for fish sizing accuracy.

In this study, we have also proved that standard IoU values are not adequate to quantify the performance of segmentation of individual fish in the overlapping situations in which specimens are occluded by other fish. A modification of the previous metric has been proposed (IoU*) as a statistic that can effectively be used for gauging the similarity of the detected masks with respect to the hand-labelled ground-truth masks.

The approach in this paper has been developed with the operational goal of achieving real-time execution on dedicated hardware inside the Deep Vision imaging system. The testing

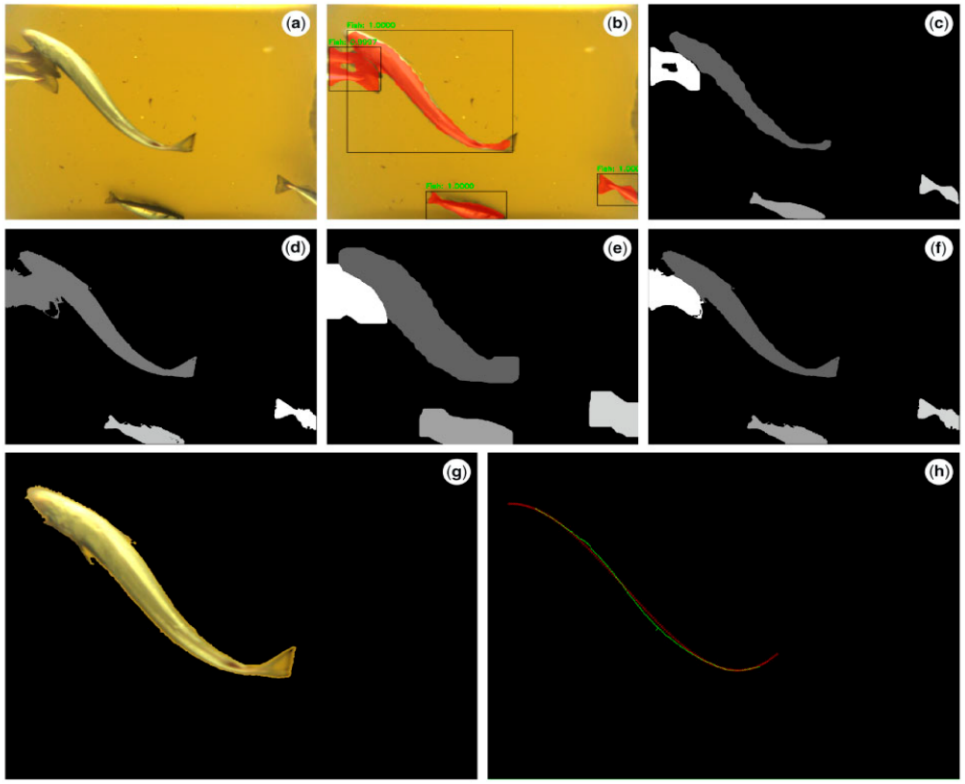


Figure 7. Automatic fish detection and length estimation. (a) Original image. (b) Fish detection and semantic segmentation through the Mask R-CNN processing. Note that the system is able to correctly detect the central fish, although it fails to detect the two tails on the left as two separate fish. (c) Labelled image as provided by Mask R-CNN. (d) Fish boundary gradient refinement mask. Note that, in this case, the segmentation is not able to distinguish among touching fish. (e) Multi-label dilate morphological operation of the Mask R-CNN segmentation. (f) Fish mask resulting of the combination of both gradient refinement and multi-label dilate. (g) Final segmented fish. (h) Skeleton pixels (in green) of the segmented fish and measurement points (in red) of the estimated fish-shape curve used to perform an automatic size measurement.

Table 3. Experiment 2: results obtained by Mask R-CNN after training with randomly selected 90% images from dataset 1 (D#1) and dataset 2 (D#2), the other 10% is reserved for testing.

		No. of images	Total no. of annotated fish	No. of detected fish with IoU* >0.7	No. of detected fish with IoU* <0.7	FN	FP	IoU*
Mask R-CNN train and valid. on 90%	Single fish	170	233	225	7	1	10	0.89
(D#1 + D#2), test in 10% (D#1 + D#2)	Overlapping fish	26	104	82	16	6	5	0.79
Gradient refinement	Single fish	170	233	224	8	1	10	0.90
	Overlapping fish	26	104	84	14	6	4	0.80

Performance taking into account the new metric IoU* that penalizes detection of a single fish when two or more fish instances are labelled in the ground-truth.

reported in this paper was conducted offline on a high-end desktop computer with a NVIDIA TITAN *V* GPU. The segmentation was run on the GPU at a frame rate was 2.67 images per second. The refinement in the current state is not optimized for speed.

A number of extensions to this work is planned in the near future. The validation of the size measurements is currently being pursuit with the intent of using fish specimens or accurate fish shape reproductions of known dimensions. The testing is to be conducted in water, to take into account the

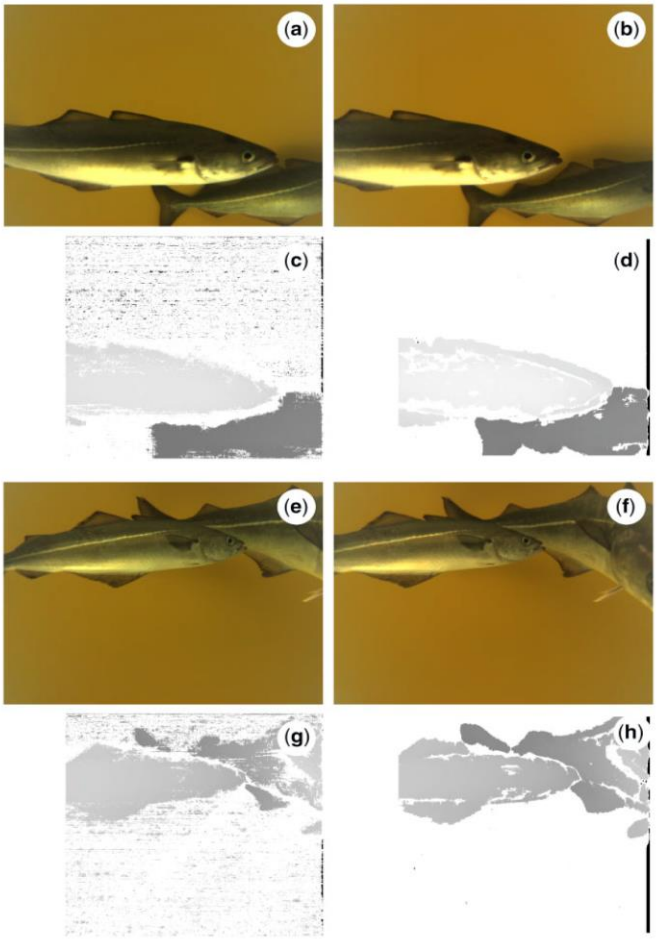


Figure 8. Traditional stereo matching techniques fail to segment overlapping fish due to lack of sufficient salient features and visual texture. The (a, b) and (e, f) images correspond to a pair of stereo images presenting fish that are partially occluded by other fish. (c, d) and (g, h) show the resulting disparity maps using two standard stereo processing techniques: (left) Semi Global Matching (Hirschmuller, 2005) and (right) Block Matching (Konolige, 1998).

refraction effects of the flat-port camera housing and how it affects the stereo geometry.

A second extension is directed towards achieving an execution frame rate in the order of 10 fps, on the target embedded processing hardware. This hardware is based on NVIDIA Jetson AGX Xavier modules and will be deployed with Deep Vision imaging system. The intended frame rate will allow performing tracking of fish across time, given that multiple instances of the same fish are likely to occur when images are acquired at 10 fps or higher, for nominal trawling speeds. This will enable the ability of estimating in real time the amount of fish in the trawl as well as the average size. Finally, as more data becomes annotated, future development will extend this work to use Mask R-CNN for automatic fish species identification.

Funding

Development of Deep Vision technology has been supported through the Research Council of Norway's Industrial PhD Programme and Innovation Norway's program for development of environmental technology (project 100424). R. Garcia and N. Gracias were partly funded by the Spanish Ministry of Education, Culture, and Sport under project CTM2017-83075-R. Data

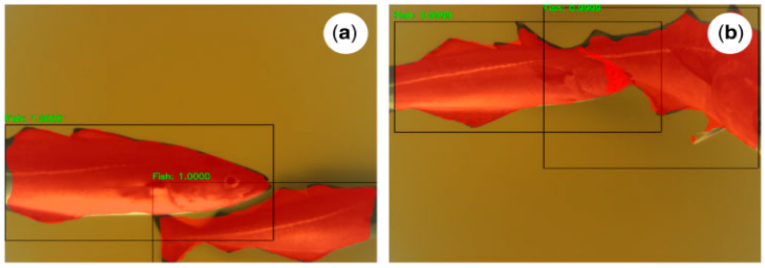


Figure 9. (a) Result of Mask R-CNN for the images of Figure 8b. (b) Instance segmentation of the two fish of Figure 8f: note the small error in the detection of the fish on the right. Although far from achieving ideal results, Mask R-CNN outperforms the state-of-the-art stereo processing techniques of Figure 8, even when the fish are not completely visible.

collection onboard R/V "*Dr* Fridtjof Nansen" was supported by the Institute of Marine Research under the CRISP centre for research innovation (Research Council of Norway project 203477) and vessel time onboard M/S "Vendla" was provided by the REDUS project with funding from the Norwegian Ministry of Trade, Industry, and Fisheries. The authors would like to thank Roger Portas for his assistance with this project.

References

Allken, V., Olav, N., Rosen, S., Schreyeck, T., Mahiout, T., and Malde, K. 2019. Fish species identification using a convolutional neural network trained on synthetic data. ICES Journal of Marine Science, 76: 342–349.

Berges, B., Sakinan, S., and van Helmond, E. 2018. Practical Implementation of Real-time Fish Classification from Acoustic Broadband Echo Sounder Data—RealFishEcho Progress Report. Wageningen Marine Research (University & Research Centre), Wageningen. Wageningen Marine Research Report, C062/18. 42 pp.

Burton, G. J. 1973. Evidence for non-linear response processes in the human visual system from measurements on the thresholds of spatial beat frequencies. Vision Research, 13: 1211–1225.

Chuang, M., Hwang, J., and Williams, K. 2016. Automatic fish segmentation and recognition for trawl-based cameras. *In* Computer Vision and Pattern Recognition in Environmental Informatics, pp. 79–106. Ed. by J. Zhou, X. Bai, and T. Caelli. IGI Global, Hershey, PA.

Costa, C., Loy, A., Cataudella, S., Davis, D., and Scardi, M. 2006. Extracting fish size using dual underwater cameras. Agricultural Engineering, 35: 218–227.

Dai, J., He, K., and Sun, J. 2016. Instance-aware semantic segmentation via multi-task network cascades. *In* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3150–3158. DOI: 10.1109/CVPR.2016.343.

Debevec, P. E., and Malik, J. 1997. Recovering high dynamic range radiance maps from photographs. *In* Proceedings of the 24th annual conference on Computer graphics and interactive techniques (SIGGRAPH '97). ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 369–378.

Dougherty, E. 1992. An Introduction to Morphological Image Processing. SPIE Optical Engineering Press. ISBN0-8194-0845-X.

FAO. 2018. The State of World Fisheries and Aquaculture. Meeting the Sustainable Development Goals, Rome, Italy. http://www.fao.org/3/i9540en/I9540EN.pdf.

Fernandes, P. G., Copland, G., Garcia, R., Nicosevici, T., and Scoulding, B. 2016. Additional evidence for fisheries acoustics: small cameras and angling gear provide tilt angle distributions and other relevant data for mackerel surveys. ICES Journal of Marine Science, 73: 8.

Fischler, M. A., and Bolles, R. C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM, 24: 381–395.

Girshick, R. 2015. Fast R-CNN. *In* IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 1440–1448. DOI: 10.1109/ICCV.2015.169.

Hartley, R., and Zisserman, A. 2003. Multiple View Geometry in Computer Vision. 2nd edn, Cambridge University Press, New York, NY.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. 2017. Mask R-CNN. *In* IEEE International Conference on Computer Vision (ICCV), Venice, pp. 2980–2988. DOI: 10.1109/ICCV.2017.322.

Hirschmuller, H. 2005. Accurate and efficient stereo processing by semi-global matching and mutual information. *In* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 807–814. IEEE, San Diego, CA.

Konolige, K. 1998. Small vision systems: hardware and implementation. *In* Proceedings of the 8th International Symposium in Robotic Research, Springer, London, pp. 203–212.

Korneliussen, R. J., Heggelund, Y., Eliassen, I. K., and Johansen, G. O. 2009. Acoustic species identification of schooling fish. ICES Journal of Marine Science, 66: 1111–1118.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2017. ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60: 84–90.

Li, Y., Qi, H., Dai, J., Ji, X., and Wei, Y. 2017. Fully convolutional instance-aware semantic segmentation. *In* The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2359–2367.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. *et al.* 2014. Microsoft COCO: common objects in context. *In* European Conference on Computer Vision (ECCV), Springer International Publishing, pp. 740–755. DOI: 10.1007/978-3-319-10602-1.

Long, J., Shelhamer, E., and Darrell, T. 2015. Fully convolutional networks for semantic segmentation. *In* IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, 2015, pp. 3431–3440. DOI: 10.1109/CVPR.2015.7298965.

Pérez, D., Ferrero, F. J., Alvarez, I., Valledor, M., and Campo, J. C. 2018. Automatic measurement of fish size using stereo vision. *In* IEEE International Instrumentation and Measurement Technology Conference (I2MTC), pp. 1–6.

Pérez Roda, M. A. (ed.), Gilman, E., Huntington, T., Kennelly, S. J., Suuronen, P., Chaloupka, M., and Medley, P. 2019. A third

assessment of global marine fisheries discards. FAO Fisheries and Aquaculture Technical Paper, 633. FAO, Rome. 78 pp.

Pobitzer, A., Ona, E., Macaulay, G., Korneliussen, R., Totland, A., Heggelund, Y., and Eliassen, I. K. 2015. Pre-catch sizing of herring and mackerel using broadband acoustics. *In* ICES Symposium on "Marine Ecosystem Acoustics (Some Acoustics)—Observing the Ocean Interior in Support of Integrated Management", pp. 25–28. Nantes, France.

Prados, R., Garcia, R., Gracias, N., Neumann, L., and Vågstøl, H. 2017. Real-time Fish Detection in Trawl Nets. *In* Proc. of the MTS/IEEE OCEANS 2017 Conference, Aberdeen, UK, pp. 1–5.

Prados, R., Garcia, R., and Neumann, L. 2014. Image Blending Techniques and Their Application in Underwater Mosaicing, Springer, Heidelberg. ISBN: 978-3-319-05557-2.

Qin, H., Li, X., Liang, J., Peng, Y., and Zhang, C. 2016. DeepFish: accurate underwater live fish recognition with a deep architecture. Neurocomputing, 187: 49–58.

Qin, H., Peng, Y., and Li, X. 2014. Foreground extraction of underwater videos via sparse and low-rank matrix decomposition. *In* ICPR Workshop on Computer Vision for Analysis of Underwater Imagery, Stockholm, 2014, pp. 65–72. DOI: 10.1109/CVAUI.2014.16.

Ren, S., He, K., Girshick, R., and Sun, J. F. 2015. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence, 39: 1137–1149.

Rosen, S., and Holst, J. C. 2013. DeepVision in-trawl imaging: sampling the water column in four dimensions. Fisheries Research, 148: 64–73.

Sanchez-Torres, G., Ceballos-Arroyo, A., and Robles-Serrano, S. 2018. Automatic measurement of fish weight and size by processing underwater hatchery images. Engineering Letters, 26: 461–472.

Schwarz, M. W., Cowan, W. B., and Beatty, J. C. 1987. An experimental comparison of RGB, YIQ, LAB, HSV, and opponent color models. ACM Transactions on Graphics, 6: 123–158.

Spampinato, C., Giordano, D., Di Salvo, R., Chen-Burger, Y.-H. J., Fisher, R. B., and Nadarajan, G. 2010. Automatic fish classification for underwater species behavior understanding. *In* Proceedings of the First ACM International Workshop on Analysis and Retrieval of Tracked Events and Motion in Imagery Streams, ACM, Firenze, Italy, pp. 45–50.

Handling editor: Cigdem Beyan