



UNIVERSIDADE D  
COIMBRA

Pedro Henrique Pinho Gomes

DETEÇÃO DA LOCALIZAÇÃO E  
ORIENTAÇÃO DE FOLÍCULOS  
CAPILARES PARA EXTRAÇÃO POR ROBÔ

Dissertação supervisionada pelos Professores Doutor Hélder de Jesus Araújo e Doutor Jorge Manuel Moreira de Campos Pereira Batista e submetida para o Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra, em cumprimento parcial dos requisitos para o Mestrado em Engenharia Eletrotécnica e de Computadores, especialização em Automação.

Setembro de 2023





UNIVERSIDADE D  
**COIMBRA**

**Deteção da localização e orientação de  
foliculos capilares para extração por  
robô**

Pedro Henrique Pinho Gomes

Coimbra, setembro de 2023





FACULDADE DE  
CIÊNCIAS E TECNOLOGIA  
UNIVERSIDADE DE  
COIMBRA

**Deteção da localização e orientação de folículos  
capilares para extração por robô**

Dissertação supervisionada pelos Professores Doutor Hélder de Jesus Araújo e Doutor Jorge Manuel Moreira de Campos Pereira Batista e submetida para o Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra, em cumprimento parcial dos requisitos para o Mestrado em Engenharia Eletrotécnica e de Computadores, especialização em Automação.

**Orientador:**

Prof. Dr. Hélder de Jesus Araújo

**Co-Orientador:**

Prof. Dr. Jorge Manuel Moreira de Campos Pereira Batista

**Júri:**

Presidente: Prof. Jorge Miguel Sá Silva

Vogal: Prof. Nuno Miguel Mendonça Gonçalves

Vogal: Prof. Hélder de Jesus Araújo

Coimbra, setembro de 2023



# Agradecimentos

Uma dissertação de mestrado representa uma jornada longa e complexa, repleta de obstáculos, momentos de tristeza, incertezas e alegrias, além de inúmeros desafios ao longo deste percurso. Apesar de ser uma experiência muitas vezes solitária, é enriquecida com as contribuições essenciais de várias pessoas, que desempenham um papel crucial ao longo desta jornada. Agradeço sinceramente a todas as pessoas que me apoiaram, forneceram energia e me deram força, pois sem elas, trilhar este caminho teria sido impossível.

Quero agradecer especialmente aos meus orientadores, Professores Doutores, Hélder Araújo e Jorge Batista, cujo apoio inabalável e disponibilidade incondicional foram fundamentais ao longo de todo este projeto. O seu compromisso e orientação foram cruciais para o sucesso deste trabalho.

Quero também agradecer a todos os membros da empresa Insparya pela sua generosidade, colaboração, parceria e tempo despendido neste projeto. Prestaram uma ajuda fundamental no processo de investigação, pois sem eles este projeto não teria sido possível.

Tendo consciência que sozinho nada disto teria sido possível, dirijo um especial agradecimento aos meus pais e irmã, por serem modelos de coragem, pelo seu apoio e amor incondicional, incentivo, amizade e paciência demonstrados e total ajuda na superação dos obstáculos que ao longo desta caminhada foram surgindo. Agradeço-lhes a enorme compreensão, generosidade e alegria com que me brindaram constantemente, contribuindo para chegar ao fim deste percurso.

A toda a minha família que sempre me apoiou e incentivou a trabalhar para que isto fosse possível. Sempre disponíveis para me ajudar quando precisei, foram também uma das grandes motivações para levar este projeto até ao fim.

À minha namorada, gostaria de expressar um agradecimento especial. A sua presença constante, compreensão infinita e apoio incondicional nos bons e maus momentos, têm sido pilares essenciais que sustentaram a minha jornada.

Quero agradecer aos meus amigos que compartilharam comigo todos estes anos em São Pedro do Sul. Os nossos momentos juntos foram repletos de gargalhadas, amizade e apoio mútuo, e isso enriqueceu a minha vida de uma maneira inestimável.

Aos amigos que conheci durante minha passagem por Coimbra, quero agradecer por terem testemunhado o meu início neste Departamento e por celebrarem comigo a minha saída. As nossas experiências são lembranças que levarei comigo para a vida.

Também gostaria de estender os meus agradecimentos à minha "família desportiva", a Associação Grupo de Ginástica de Vouzela. O apoio e camaradagem que encontrei neste grupo foram cruciais para minha jornada e crescimento pessoal. Sem a comunidade que vocês representam, não teria alcançado tudo o que alcancei.

A todos vós, a minha mais profunda gratidão. Sem o apoio, amor e amizade de cada um, as minhas conquistas não seriam possíveis. Vocês são a base que sustenta o meu sucesso, e por isso sou eternamente grato.



# Abstract

Hair loss is a problem that affects numerous people worldwide, regardless of their age or gender. Due to various factors, including stress, hormonal imbalances, or genetics, this medical condition is significantly affecting individuals' self-esteem and quality of life. This past decade, it has become increasingly common for people who had been suffering from hair loss or alopecia to seek a solution to correct this issue. This is where hair transplants appear as the most popular option, providing a more natural and dense appearance. This dissertation aims to provide a solution to improve the effectiveness of the Follicular Unit Extraction (FUE) method. This technique involves extracting follicular units from donor areas with higher density to be replanted in bald or thinning areas. Our goal is to enhance the follicle detection performance along with their orientation using neural networks. There are several pre-trained networks for object detection, which, in this specific case, would be follicles. However, the main challenge lies in detecting the follicular unit with its proper orientation, represented as pose within a 2D image. Once this objective is achieved, it would be beneficial to track the detected follicles across frames captured by a camera attached to a robotic arm. This tracking would assist in identifying each follicle across frames for subsequent extraction. This work has been developed in partnership with Insparya, a leading company in what concerns the hair transplant market, known for investing in innovative and advanced technologies to enhance the accuracy and efficiency of the process. Currently, traditional Computer Vision methods are being used to detect follicular units, improving detection precision and speed. Real-time detection necessitates minimizing the detection time to prevent interference with any additional calculations that might be performed during the process.

The whole work will be implemented in ROS[1] to ensure easy integration into the company's software. ROS is highly advantageous for robotics since it facilitates communication between different aspects of a project, both hardware and software.

*Keywords:* Oriented Object Detection, Neural Networks, Tracking, DeepLearning, ROS



# Resumo

A queda de cabelo é um problema que afeta diversas pessoas em todo o mundo, independentemente da sua idade ou género. Esta pode ser causada por vários fatores, entre eles, stresse, desequilíbrios hormonais ou genética, acabando por afetar significativamente a autoestima e a qualidade de vida das pessoas.

Na última década, tornou-se cada vez mais comum as pessoas que sofrem de queda de cabelo ou alopecia procurarem uma solução capaz de corrigir este problema. Os transplantes capilares são uma opção popular, proporcionando uma aparência mais natural e densa.

Esta dissertação vai ao encontro de uma solução para a melhoria da eficácia do método *Follicular Unit Extraction* (FUE), técnica esta que consiste em extrair folículos em zonas doadoras e mais densas, para que sejam novamente plantados na zona calva ou com menos cabelo. O objetivo deste trabalho é melhorar a performance de deteção de folículos com a sua orientação através de redes neuronais. Existem diversas redes já pré-treinadas para fazer a deteção de objetos, que no caso seriam folículos, mas o maior desafio será fazer a deteção da unidade folicular com a sua devida orientação, ou seja, a sua pose numa imagem 2D.

Após este objetivo ser conseguido, seria benéfico fazer o *tracking* dos mesmos folículos detetados ao longo dos *frames* captados por uma câmara acoplada a um braço robótico, visto que ajudaria a identificar cada folículo entre *frames* para a sua posterior extração.

Este trabalho foi desenvolvido em conjunto com a empresa *Insparya*, atualmente líder no mercado de transplantes capilares e tem-se destacado por investir em tecnologias inovadoras para melhorar a precisão e eficiência do processo. Neste momento, estão a ser utilizados métodos tradicionais de "Visão por Computador" para ser feita a deteção de unidades foliculares, melhorando a deteção tanto a nível de precisão, como de rapidez. A deteção em tempo real implica que o tempo de deteção seja o menor possível para que, caso hajam cálculos extra a serem efetuados, estes não interfiram com o tempo de todo o processo.

Todo este trabalho será implementado em ROS[1] de modo a que seja facilmente integrado no software da empresa. ROS é extremamente benéfico em tudo o que envolve robótica, pois facilita a comunicação entre os diferentes aspetos de um projeto tanto hardware como software.

*Palavras-chave:* Deteção Orientada de Objetos, Redes Neuronais, Tracking, DeepLearning, ROS



*“A persistência é o menor caminho do êxito”*

Charles Chaplin



# Conteúdo

<b>Acknowledgments</b>	<b>i</b>
<b>Abstract</b>	<b>iv</b>
<b>Resumo</b>	<b>vi</b>
<b>List of Acronyms</b>	<b>xiii</b>
<b>Lista de Figuras</b>	<b>xv</b>
<b>Lista de Tabelas</b>	<b>xvii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contexto e Motivação . . . . .	1
1.2 Objetivos . . . . .	2
1.3 Metodologia . . . . .	2
<b>2 Trabalho Realizado e Contribuições</b>	<b>4</b>
<b>3 Revisão da Literatura</b>	<b>6</b>
3.1 Técnicas de transplantes capilares . . . . .	6
3.1.1 FUE - Extração de Unidade Folicular . . . . .	6
3.1.2 FUT - Transplante de Unidade Folicular . . . . .	7
3.1.3 FUT vs FUE . . . . .	7
3.2 Robô ARTAS . . . . .	8
3.3 KEBOT . . . . .	10
3.3.1 Sistema de análise baseado em Inteligência Artificial . . . . .	12
3.3.1.1 Detecção de objetos . . . . .	12
3.3.1.2 Cálculo da espessura do folículo . . . . .	13
3.3.1.3 Análise Métrica . . . . .	13
3.4 Métodos tradicionais de VPC . . . . .	14
3.4.1 Segmentação de Unidades Foliculares . . . . .	14
3.5 Redes Neurais . . . . .	15
3.5.1 YOLO: "You Only Look Once" . . . . .	15
3.5.2 R-CNN: "Regions with CNN feature & Fast/Faster-RCN" . . . . .	15

<b>4</b>	<b>Trabalho Desenvolvido</b>	<b>18</b>
4.1	Data . . . . .	18
4.1.1	Verificação de Imagem . . . . .	18
4.1.2	Conversão de Formato das <i>labels</i> . . . . .	19
4.1.3	Anotações de Imagens . . . . .	22
4.2	Cálculo das métricas . . . . .	23
4.3	Redes Neurais Sem Orientação . . . . .	24
4.3.1	<i>YOLOv5</i> . . . . .	24
4.3.2	<i>YOLOv8</i> . . . . .	29
4.4	Teste <i>Tracking</i> . . . . .	34
4.5	Rede detecção Orientada - <i>YOLOv5_OBB</i> . . . . .	36
4.5.1	Introdução à rede <i>YOLOv5_OBB</i> . . . . .	36
4.5.2	Treinos Iniciais da Rede <i>YOLOv5_OBB</i> . . . . .	38
4.5.3	Tempos de Computação . . . . .	40
4.5.4	Melhoramento do <i>dataset</i> (Novos Treinos) . . . . .	42
4.5.4.1	Métricas Empresa . . . . .	46
4.6	Recriação 3D . . . . .	49
4.7	<i>Tracking</i> com a detecção orientada . . . . .	50
4.8	Interfaces . . . . .	52
<b>5</b>	<b>Conclusão</b>	<b>55</b>
5.1	Comparação Redes . . . . .	55
5.2	Tracking Adicionado . . . . .	57
5.3	Trabalho Futuro . . . . .	58
5.3.1	Expansão do Conjunto de Dados . . . . .	58
5.3.2	Adicionar Novas Classes . . . . .	58
5.3.3	Otimizar Tempos de Detecção . . . . .	58
5.3.4	Treinos Com Maior Poder Computacional . . . . .	58
5.3.5	Explorar Novas Redes Neurais e Diferentes Técnicas de Detecção . . . . .	58
	<b>Bibliography</b>	<b>60</b>



# Lista de Acrónimos

- IoU** Intersection over Union
- FUE** Follicular Unit Extraction
- FUT** Follicular Unit Transplantation
- YOLO** You Only Look Once
- R-CNN** Regions with CNN feature
- SORT** Simple Online and Realtime Tracking
- ROS** Robot Operating System
- RQT** ROS Qt-based User Interface Toolkit
- VPC** Visão Por Computador
- CSL** Circular Smooth Label
- mAP** mean Average Precision
- NMS** Non-Maximum Suppression
- TP** True Positive
- TN** True Negative
- FP** False Positive
- FN** False Negative
- UC** Universidade de Coimbra



# Lista de Figuras

3.1	Imagem do sistema <i>ARTAS</i> retirada de [2]. . . . .	9
3.2	Imagem do sistema <i>KEBOT</i> retirada de [3]. . . . .	10
3.3	Diagrama do sistema <i>KEBOT</i> retirada de [3]. . . . .	11
3.4	<i>KEBOT</i> object detection, imagem retirada de [3]. . . . .	12
4.1	Imagens Mal Anotadas/Desfocadas. . . . .	19
4.2	Transformação formato de <i>labels</i> . . . . .	20
4.3	Transformação formato de <i>labels</i> (Exemplo 1). . . . .	21
4.4	Transformação formato de <i>labels</i> . (Exemplo 2) . . . . .	21
4.5	Extrapolção da Imagem . . . . .	22
4.6	Programa de Anotações Desenvolvido pela Empresa . . . . .	23
4.7	Estrutura do Modelo <i>YOLOv5</i> [4]. . . . .	25
4.8	Evolução da Detecção de Unidades Foliculares . . . . .	26
4.9	Métricas do Treino da Rede <i>YOLOv5</i> . . . . .	27
4.10	Matriz de Confusão da Rede <i>YOLOv5</i> . . . . .	27
4.11	Resultados da Rede <i>YOLOv5</i> . . . . .	28
4.12	Comparação de unidades foliculares detetadas pela rede <i>YOLO</i> (Azul) e pelo software da empresa <i>Insparya</i> (Verde) . . . . .	29
4.13	Estrutura da rede <i>YOLOv8</i> retirado de [5]. . . . .	30
4.14	Comparação versões <i>YOLO</i> relativamente ao <i>mAP</i> e <i>Latency</i> [5]. . . . .	31
4.15	Detecções <i>YOLOv8</i> a Preto e Branco . . . . .	32
4.16	Detecções <i>YOLOv8</i> a Cores . . . . .	32
4.17	Métricas do Treino da Rede <i>YOLOv8</i> . . . . .	33
4.18	Matriz de Confusão da Rede <i>YOLOv8</i> . . . . .	33
4.19	Resultados da Rede <i>YOLOv8</i> . . . . .	34
4.20	<i>Tracking</i> Output através das redes <i>YOLOv5</i> e <i>YOLOv8</i> . . . . .	35
4.21	Ilustração para o problema de fronteira que persistentemente existe em 3 categorias populares de protocolos baseados em regressão. A seta vermelha sólida indica o processo de regressão real, e a tracejada vermelha mostra o processo de regressão ideal.[6] . . . . .	36
4.22	Dois tipos de <i>labels</i> para classificação angular.[6] . . . . .	37
4.23	Treinos Iniciais da Rede. . . . .	38
4.24	Treinos Finais da Rede. . . . .	39

4.25	Comparação Tempos de Deteção. . . . .	40
4.26	Hiper-Parâmetros utilizados no Treino 7 da Tabela 4.7. . . . .	44
4.27	Deteções efetuadas com o Treino 7 da Tabela 4.7. . . . .	44
4.28	Matriz de Confusão e Correlograma das Labels Provenientes do Treino 7 da Rede YOLOv5_OBB. . . . .	45
4.29	Curvas das Métricas Provenientes do Treino 7 da Rede YOLOv5_OBB. . . . .	46
4.30	Resultados Provenientes do Treino 7 da Rede YOLOv5_OBB. . . . .	46
4.31	Métricas Obtidas Através das Deteções Efetuadas Pelo Software da Empresa . .	48
4.32	Exemplo Deteção com Software de Deteção da Empresa Insparya . . . . .	48
4.33	Exemplo Deteção com Software de Deteção da Empresa Insparya . . . . .	49
4.34	Exemplo IoU. . . . .	50
4.35	Tracking Utilizando a Rede YOLOv5_OBB. . . . .	51
4.36	Interface RQT. . . . .	53

# Lista de Tabelas

4.1	Métricas Derivadas do Treino da Rede <i>YOLOv5</i> . . . . .	28
4.2	Métricas Derivadas do Treino da Rede <i>YOLOv8</i> . . . . .	34
4.3	Parâmetros e Resultados Iniciais . . . . .	39
4.4	Tempos de Inferência da rede <i>YOLOv5_OBB</i> em segundos . . . . .	41
4.5	Tabela de Comparação dos Tempos de Detecção Total das Imagens em segundos .	41
4.6	Tabela das Médias de Comparação dos Tempos de Detecção das Imagens em segundos	42
4.7	Treinos com <i>dataset</i> completo . . . . .	43
5.1	Comparação de <i>Precision</i> e <i>Recall</i> para diferentes valores de IOU Threshold com o melhor Treino da Rede. . . . .	56
5.2	Comparação de Vantagens e Desvantagens entre Rede neuronal e Métodos Tradicionais de VPC. . . . .	57



# 1

## Introdução

### 1.1 Contexto e Motivação

O folículo piloso é tipicamente observado nos mamíferos, conferindo funções protetoras, termorreguladoras e sensoriais. A nível protetor, a detenção de pelo funciona como escudo da pele, seja contra danos físicos imediatos, como picadas de insetos, seja contra o sol e o frio. Mantém também a temperatura corporal estável, colaborando com a termorregulação.

A alopecia é uma condição que causa a perda parcial ou total de cabelo, que apresenta impacto na vida de pessoas, independentemente do seu género ou da sua idade. Quando se fala de folículos capilares, é necessário que se entenda, também, a sua função psicossocial e o peso que certas anomalias podem apresentar na vida dos indivíduos, principalmente a nível emocional e a nível fisiológico. Diversas técnicas de transplante capilar foram desenvolvidas para combater a alopecia, sendo a extração de unidade folicular (FUE) uma das técnicas mais populares. A FUE é uma técnica que envolve a extração de folículos individuais do couro cabeludo sem deixar cicatrizes visíveis.

Nos últimos anos, o uso de robôs para realizar a extração de folículos capilares tem-se mostrado uma opção promissora para melhorar a precisão e a eficiência do processo. No entanto, a deteção da localização e orientação precisa dos folículos capilares é um desafio técnico importante na utilização de robôs para extração de folículos capilares.

Esta tese de mestrado tem como objetivo desenvolver uma técnica para a deteção da localização e orientação de folículos capilares para extração por robô. Para alcançar este objetivo, serão exploradas técnicas de processamento de imagens, inteligência artificial e robótica para desenvolver um sistema de deteção precisa e eficiente.

Os resultados desta pesquisa serão importantes para melhorar a eficácia e segurança da extração de folículos capilares por robôs, permitindo resultados estéticos superiores e reduzindo o tempo de recuperação dos pacientes. Esta pesquisa tem o potencial de contribuir significativamente para a área de medicina estética e para a melhoria da qualidade de vida de pacientes com alopecia.

A empresa *Insparya* é líder no mercado de transplante capilar e tem-se destacado por investir em tecnologias avançadas para melhorar a precisão e eficiência do processo. Este projeto tem como objetivo, trabalhar com esta empresa, de modo a tentar encontrar uma solução mais eficaz

para o processo de localização das unidades foliculares.

## 1.2 Objetivos

O objetivo principal neste trabalho, é o desenvolvimento de um estudo sobre o uso de redes neuronais para substituir os métodos tradicionais de Visão por Computador na detecção de unidades foliculares.

Existem diversas redes neuronais para fazer a detecção de objetos e sendo necessária a orientação da unidade folicular, poderia ser uma opção utilizar estas redes para a detecção das unidades foliculares e dentro de cada *bounding box* detetada determinar a sua orientação. O problema disto é que necessitaria de uma grande capacidade computacional e iria consumir demasiado tempo, o que é crucial neste projeto.

Sendo assim, teve que se utilizar uma rede, a qual seria treinada, fornecendo diretamente a detecção e orientação das unidades foliculares.

Neste sentido e após ter sido cumprido esse objetivo, seria também benéfico fazer o *tracking* dos folículos detetados, para ser possível serem distinguidos entre os diferentes *frames* de imagens captados por uma câmara acoplada a um braço robótico.

Após a rede neuronal e o *tracking* estarem funcionais, será necessário fazer a sua implementação em ROS, para uma fácil comunicação e integração direta no software da empresa.

## 1.3 Metodologia

A criação de um *dataset* é uma parte crucial no contexto do treino de redes neuronais, mas isto implica obter as imagens, anotá-las, dividir em conjuntos de treino, validação e teste, e realizar o pré-processamento das imagens. O conjunto de dados bem construído é fundamental para garantir a qualidade dos resultados obtidos. Além disso, é importante destacar a importância do melhoramento desse *dataset* para ter uma estrutura mais robusta e melhorar a precisão da detecção de folículos.

Estes dados anotados serão utilizados para treinar as redes *YOLOv5*[4] e *YOLOv5\_OBB*[7], permitindo a detecção e classificação de unidades foliculares em imagens do couro cabeludo de pacientes. Primeiramente será iniciado o treino da rede *YOLOv5* para ver como se comporta em relação ao *dataset* obtido previamente. Posteriormente, quando forem obtidos alguns resultados, será alterado o formato das *labels* para a rede *YOLOv5\_OBB* que fornece diretamente a orientação, no entanto, precisa de um treino ligeiramente diferente.

Para melhorar a qualidade do conjunto de dados, é fundamental aumentar o número de imagens e incluir variações nas condições de imagem, como diferentes iluminações, orientações, cores, entre outros. Também é importante incluir diferentes tipos de folículos, considerando diferentes tamanhos e formatos. Técnicas de "*data augmentation*", como a rotação e o *flip* das imagens, também devem ser aplicadas para aumentar a quantidade de imagens no conjunto de dados e tornar a rede neural mais robusta em relação a variações nas condições de imagem. Isto

será um processo que decorrerá desde o início até ao fim, um melhoramento constante do *dataset* para a maior robustez possível alcançada.

Em seguida, será aplicado o algoritmo *SORT* (*"Simple Online and Realtime Tracking"*)[8] para rastrear os folículos detetados em tempo real, permitindo o acompanhamento do trajeto dos folículos ao longo do tempo.

Para integrar todos esses processos, será utilizada a estrutura *ROS* (*"Robot Operating System"*), que é uma plataforma de software utilizada para desenvolver sistemas robóticos complexos. Isso permitirá a implementação de todos os processos em um único sistema, facilitando a interação entre as diferentes partes da *framework*.

# 2

## Trabalho Realizado e Contribuições

Este projeto é uma colaboração com a empresa Insparya, líder no campo da saúde capilar. A Insparya já realizou um trabalho significativo no desenvolvimento de métodos de detecção de unidades foliculares, gotas de sangue e orifícios deixados pelo transplante capilar, além de técnicas de segmentação e *tracking* de folículos.

A detecção de unidades foliculares é um aspecto fundamental no campo da saúde capilar e a Insparya tem-se dedicado a melhorar essa etapa crucial. Desenvolveram métodos inovadores utilizando técnicas tradicionais de Visão por Computador (VPC). Além disso, a empresa tem aplicado técnicas de segmentação para identificar os orifícios deixados pelo transplante capilar, permitindo uma avaliação mais precisa da eficácia do procedimento.

Outro aspecto importante abordado pela empresa Insparya é o *tracking* de folículos ao longo do tempo. O acompanhamento preciso do crescimento e movimento dos folículos é essencial para monitorar o progresso do tratamento capilar. Através de métodos tradicionais de VPC, a empresa conseguiu rastrear e estimar a trajetória dos folículos com precisão. Isso fornece informações valiosas para a avaliação da eficácia do tratamento e a tomada de decisões clínicas.

Como parte deste projeto de colaboração, será utilizado algum do trabalho já realizado pela empresa e será fundamental ter acesso a partes relevantes do software, do qual será extraído o *dataset*, que contém imagens e informações cruciais para a nossa pesquisa. Esses dados serão fundamentais para treinar e validar a rede neural, permitindo que ela aprenda a reconhecer e analisar unidades foliculares com precisão.

Além disso, foi aproveitada a interface do ambiente em ROS criada pela Insparya. O ROS ("*Robot Operating System*") é uma *framework* de código aberto amplamente utilizado na área de robótica e visão computacional. A Insparya implementou todo o seu software em ROS, proporcionando uma base sólida para integrar essas contribuições. Utilizando a interface fornecida, serão adicionados os algoritmos criados e as técnicas de redes neurais, enriquecendo ainda mais a capacidade de detecção e *tracking* de folículos capilares.



# 3

## Revisão da Literatura

### 3.1 Técnicas de transplantes capilares

As técnicas de transplante capilare são procedimentos avançados que oferecem soluções eficazes para combater a perda de cabelo. Entre as mais comuns estão, a Extração de Unidade Folicular (FUE) e o Transplante de Unidade Folicular (FUT). No entanto, este trabalho concentra-se especificamente na técnica FUE, procurando otimizar e automatizar ao máximo esse processo. Esta abordagem visa combinar a experiência dos profissionais com o uso de tecnologia avançada, para oferecer resultados mais precisos e eficazes.

#### 3.1.1 FUE - Extração de Unidade Folicular

Relativamente à Extração de Unidade Folicular (FUE) é uma técnica de transplante capilar minimamente invasiva e avançada. Nesse procedimento, os folículos capilares são extraídos individualmente da área doadora, que geralmente é a parte de trás ou dos lados do couro cabeludo, onde o cabelo é geneticamente resistente à queda.

Os principais passos do procedimento FUE são:

- Preparação da área doadora: A área doadora é preparada e anestesiada localmente para minimizar qualquer desconforto durante o procedimento.
- Extração dos folículos: Com o uso de uma pequena ferramenta chamada "punch", o cirurgião realiza pequenas incisões circulares ao redor de cada folículo capilar individual. Essas incisões podem variar em diâmetro, dependendo das características dos folículos a serem extraídos. Após a criação das incisões, os folículos capilares são cuidadosamente retirados. É importante lembrar que essa etapa requer um esforço do cirurgião, que necessita da precisão e estabilidade necessária para não causar danos nos folículos e até mesmo no paciente.
- Preparação da área recetora: Enquanto os folículos estão a ser extraídos, a área recetora, onde os folículos serão implantados, é preparada através de pequenas incisões ou microcanais.
- Implantação dos folículos: Os folículos capilares extraídos são então cuidadosamente implantados nas incisões ou microcanais na área recetora. O cirurgião coloca os folículos

numa direção e padrão de crescimento cuidadosamente planejados para garantir resultados estéticos naturais.

É importante ressaltar que o procedimento FUE é um processo bastante demorado, podendo levar várias horas ou até mesmo dias, dependendo da extensão do transplante e do número de folículos a serem extraídos e implantados. Além disso, o sucesso do procedimento depende da habilidade e experiência do cirurgião, bem como de uma avaliação completa do paciente para determinar a adequação do transplante capilar.[9, 10, 11, 12]

### 3.1.2 FUT - Transplante de Unidade Folicular

Relativamente ao Transplante de Unidade Folicular (FUT), é uma técnica de transplante capilar tradicional e eficaz. Neste procedimento, uma faixa de tecido contendo folículos capilares é removida da área doadora, geralmente na parte de trás ou dos lados do couro cabeludo, onde o cabelo é geneticamente mais resistente à queda.

Os principais passos do procedimento FUT são:

- Preparação da área doadora: A área doadora é preparada e anestesiada localmente para minimizar qualquer desconforto durante o procedimento.
- Remoção da faixa de tecido: Com o uso de técnicas cirúrgicas precisas, uma faixa de tecido contendo folículos capilares é removida da área doadora. Essa faixa é então cuidadosamente suturada para fechar a incisão.
- Dissecção dos folículos: A faixa de tecido removida é cuidadosamente dividida em unidades foliculares individuais, que consistem em um ou mais folículos capilares.
- Preparação da área recetora: Enquanto os folículos são dissecados, a área recetora, tal como na técnica mencionada anteriormente (FUE), são criados microcanais ou pequenas incisões para que possam implantar os folículos.
- Implantação dos folículos: Os folículos capilares individuais são então cuidadosamente implantados nas incisões ou microcanais na área recetora, seguindo um padrão de distribuição cuidadosamente planejado para obter resultados estéticos naturais.

Temos que ter em conta que o tempo de recuperação é normalmente mais longo em comparação com a técnica FUE e a cicatriz resultante da remoção da faixa de tecido será visível, podendo ser disfarçada pelo crescimento capilar envolvente.[9, 13]

### 3.1.3 FUT vs FUE

Ainda que as duas técnicas supracitadas sejam eficazes, estas são aplicadas a casos diferentes, apresentando algumas diferenças significativas.

O FUT envolve a remoção de uma faixa de tecido da área doadora, geralmente na parte de trás ou dos lados do couro cabeludo, onde o cabelo é geneticamente resistente à queda. Essa faixa é então dividida em unidades foliculares individuais, que são implantadas na área recetora.

No entanto, esta técnica permite o transplante de um grande número de folículos de uma única vez, fazendo com que este procedimento deixe uma cicatriz linear visível na área doadora.

O FUE - por outro lado - é um método menos invasivo. Através dele, os folículos capilares são extraídos individualmente da área doadora e essas unidades foliculares são, conseqüentemente, implantadas na área recetora. Para além de ser uma abordagem menos agressiva para o paciente, este método permite, então, uma maior flexibilidade a nível de design do cabelo, visto que os folículos podem ser direcionados para áreas específicas. A grande vantagem do FUE é não deixar uma cicatriz linear visível, tornando-o preferido por muitos pacientes.[9, 11]

As duas técnicas têm as respetivas vantagens e desvantagens. O FUT é frequentemente recomendado para pacientes que precisam de um grande número de folículos transplantados, enquanto o FUE é mais adequado para casos em que a preocupação com a cicatriz é uma prioridade ou quando a flexibilidade na distribuição dos folículos é desejada.

	<b>FUT (Transplante de Unidade Folicular)</b>	<b>FUE (Extração de Unidade Folicular)</b>
<b>Vantagens</b>	<ul style="list-style-type: none"><li>- Permite o transplante de um grande número de folículos de uma única vez.</li><li>- Pode ser mais adequado para pacientes que precisam de um grande número de folículos transplantados.</li></ul>	<ul style="list-style-type: none"><li>- Não deixa uma cicatriz linear visível na área doadora.</li><li>- Maior flexibilidade no design do cabelo, pois os folículos podem ser direcionados para áreas específicas.</li></ul>
<b>Desvantagens</b>	<ul style="list-style-type: none"><li>- Deixa uma cicatriz linear visível na área doadora.</li><li>- Tempo de recuperação pós-operatório mais longo.</li><li>- Restrições na escolha do penteado do paciente, devido à cicatriz visível.</li></ul>	<ul style="list-style-type: none"><li>- Processo demorado que pode levar várias horas ou até mesmo dias.</li><li>- Necessita de um cirurgião habilitado e experiente.</li><li>- Pode ser menos adequado para transplantes que requerem um grande número de folículos.</li><li>- Pode resultar em um maior custo do procedimento.</li></ul>

A escolha entre FUT e FUE depende das necessidades e preferências individuais do paciente, bem como das características do couro cabeludo. É essencial consultar um cirurgião especializado em implantes capilares para determinar a técnica mais adequada para cada caso específico.

### **3.2 Robô ARTAS**

O robô *ARTAS* é um sistema avançado de restauração capilar como o que está representado na Figura 3.1 que utiliza tecnologia robótica para auxiliar em procedimentos de transplante de cabelo[2]. Este é um projeto desenvolvido pela empresa *Restoration Robotics*, sendo um exemplo notável em como a robótica está a revolucionar a medicina estética e oferecendo soluções inovadoras para a queda de cabelo.



**Figure 3.1:** Imagem do sistema *ARTAS* retirada de [2].

A arquitetura do robô *ARTAS* é projetada para garantir precisão e eficiência durante o procedimento de transplante capilar.

O sistema *ARTAS* é equipado com um conjunto de câmaras e sensores de imagem, o que permite ao robô mapear e digitalizar o couro cabeludo do paciente com alta resolução. Essa capacidade de imagem ajuda a identificar os melhores folículos capilares a serem extraídos e posteriormente implantados na zona calva, garantindo um resultado mais natural.

Uma das suas funções principais é a extração de unidades foliculares. Com base nas informações recolhidas pelas câmaras e sensores, o robô é capaz de identificar e isolar folículos individuais de maneira precisa. Em seguida, utiliza um pequeno instrumento de punção para extrair os folículos selecionados sem a necessidade de incisões manuais. Essa abordagem é minimamente invasiva, reduzindo o desconforto e o tempo de recuperação pós-operatório do paciente.

Além disso, também desempenha um papel fundamental na etapa de implantação dos folículos capilares extraídos. O robô utiliza uma agulha especializada para fazer pequenos orifícios no couro cabeludo do paciente, nos quais os folículos são cuidadosamente inseridos. A sua precisão e consistência garantem um enxerto capilar uniforme e natural, evitando danos nos folículos adjacentes. É também utilizado como uma ferramenta de diagnóstico para ajudar os médicos a avaliar a saúde e a qualidade dos folículos capilares. As informações detalhadas fornecidas pelo robô permitem que os médicos personalizem os planos de tratamento para cada paciente, levando em consideração fatores como a densidade capilar existente e a disponibilidade de folículos doadores.

Foi realizado um estudo com 3 pacientes consecutivos considerados candidatos ideais, após excluir problemas no couro cabeludo. Foram observados os 3 primeiros grupos de colheita de cada paciente, com um número de tentativas de colheita variando de 128 a 151 por grupo. Avaliou-se visualmente a quantidade de enxertos prontos para extração após cada tentativa de colheita feita pelo robô. Em média, registaram-se lacunas em 5,8% das áreas de colheita, onde nenhum enxerto foi visível. O robô *ARTAS* possui um depósito para enxertos que foram colhidos em excesso e sugados pelo sistema, o que resolve o problema de enxertos perdidos e lacunas. Detetou-se que, aproximadamente 40% dessas lacunas continham enxertos recuperáveis

na câmara. Após recuperar esses enxertos, constatou-se que a média real de enxertos ausentes foi de 2,4%, representando as lacunas sem enxertos recuperáveis.

Isto tem implicações significativas, sendo que a maioria dos dispositivos de FUE é controlada manualmente. Como por exemplo, um transplante capilar de 1000 enxertos, se for descoberto que durante a contagem, apenas 900 são de qualidade, é necessária a extração mais 100 enxertos. O único problema, é o tempo extra necessário que vai acrescentar à equipa médica, aumentando o stress da mesma e aumentando também o tempo que os enxertos permanecem fora do corpo após a colheita inicial.

Neste aspeto, o Artas é crucial devido à sua avaliação *à priori* da qualidade dos enxertos e da sua rapidez de extração.

O avanço da tecnologia robótica está a trazer perspectivas promissoras para a produção de transplantes capilares baseados em FUE com alta qualidade.[14]

### 3.3 KEBOT

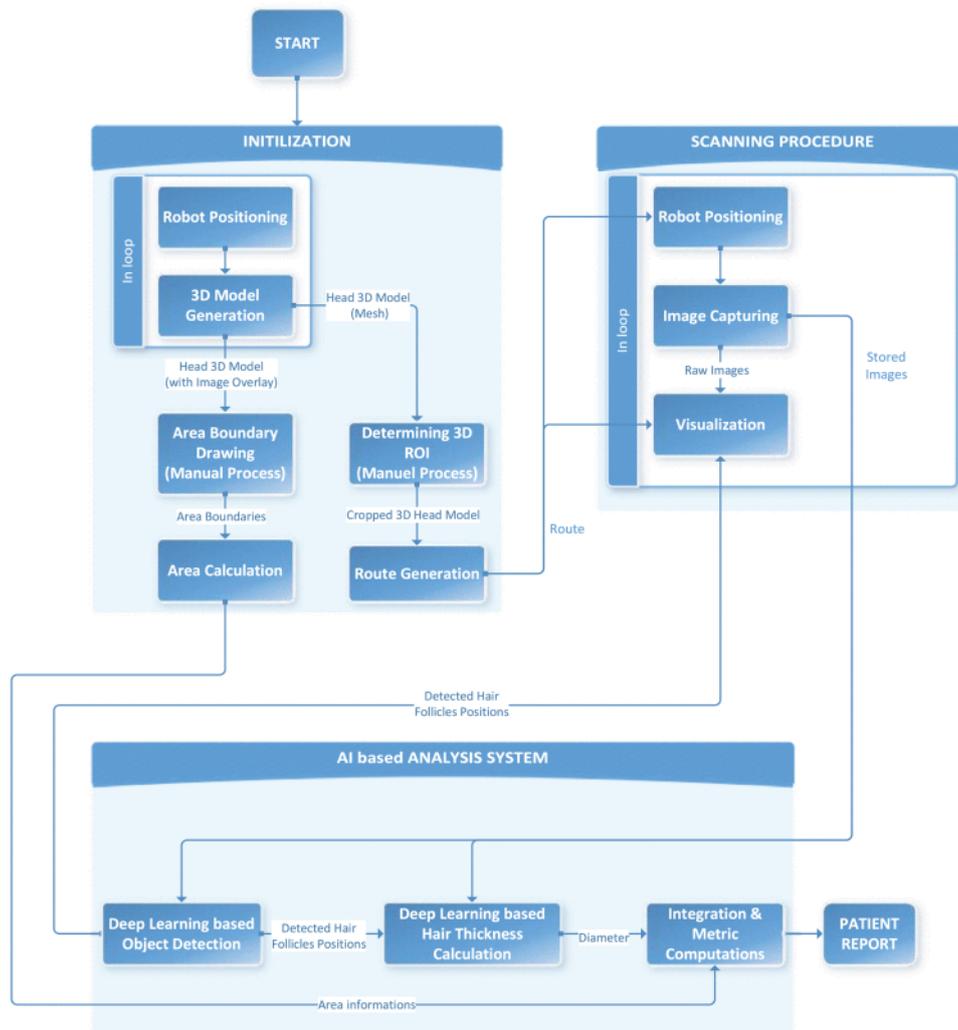
O *KEBOT*[3] é um sistema robótico avançado usado tanto na análise pré-operatória, quanto na pós-operatória de um transplante capilar. Na fase pré-operatória, o sistema deteta os folículos presentes e calcula a sua espessura, fornecendo informações detalhadas sobre a capacidade de doação do paciente. Já na fase pós-operatória, o *KEBOT* determina quantos folículos foram extraídos da área doadora e quantos foram implantados na área recetora, permitindo avaliar o desempenho do cirurgião, através da identificação de possíveis transecções ou desperdícios de folículos.



**Figure 3.2:** Imagem do sistema *KEBOT* retirada de [3].

Na Figura 3.2 está representado todo o sistema *KEBOT*. Este utiliza um braço robótico colaborativo de 6 eixos para posicionar as câmaras de profundidade e RGB, que capturam imagens em alta resolução da cabeça do paciente. A câmara de profundidade cria modelos 3D do paciente, enquanto a câmara RGB deteta os folículos pilosos e regista os folículos extraídos e implantados.

O processo de análise é dividido em 3 etapas principais: inicialização, varredura e análise. Na etapa de inicialização, é gerado um modelo 3D do paciente e, na etapa de varredura, as imagens em alta resolução são capturadas seguindo uma rota pré-determinada. Os dados 3D são utilizados para calcular a área das regiões relevantes para o transplante. Na etapa de análise, é utilizada *deep learning* nas imagens para detetar os folículos, orifícios (locais de extração) e enxertos implantados. Em seguida, os dados são processados para calcular a espessura dos fios e gerar um relatório completo sobre o procedimento.



**Figure 3.3:** Diagrama do sistema *KEBOT* retirada de [3].

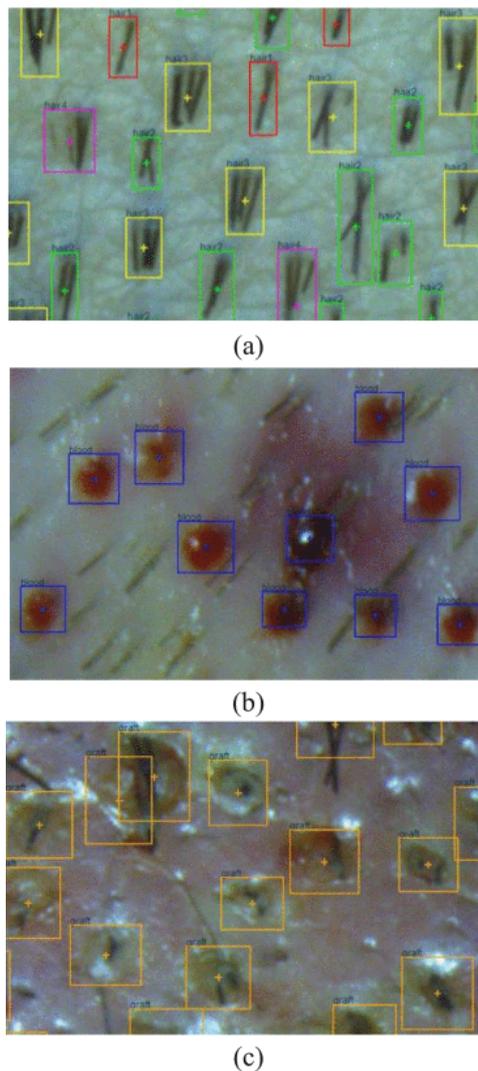
O diagrama apresentado na Figura 3.3 mostra-nos todo o processo do sistema *KEBOT*. Todas as conexões entre software utilizadas. Toda a parte baseada em Inteligência Artificial a comunicar com o restante sistema.

### 3.3.1 Sistema de análise baseado em Inteligência Artificial

O sistema proposto de análise capilar baseado em *deep learning* compõe-se em três etapas principais: detecção de objetos, cálculo de espessura do cabelo e análise métrica. Estas etapas são combinadas para fornecer resultados precisos.

#### 3.3.1.1 Deteção de objetos

No sistema *KEBOT*, a detecção e contagem de folículos capilares, enxertos pós-operatórios e vestígios de cicatrizes são tratados como um problema de detecção de objetos. Para resolver esse problema, é utilizada uma abordagem baseada em *deep learning*.



**Figure 3.4:** *KEBOT* object detection, imagem retirada de [3].

Na Figura 3.4 estão representadas as diferentes utilizações das redes neurais, em (a) temos a identificação com a devida classificação das unidades foliculares, com 1, 2 ou 3 folículos no mesmo orifício, em (b) temos a detecção de sangramentos provenientes da extração folicular, e em (c) a detecção dos folículos implantados já na nova zona.

O sistema *KEBOT* recolhe imagens em alta resolução dos pacientes durante a varredura para

um posterior treino e teste de uma rede neuronal. Milhares de imagens pré e pós-operatórias são recolhidas de pacientes reais e são manualmente anotadas por enfermeiros especializados em transplante capilar. Em cada imagem, existem aproximadamente de 250 a 450 amostras a serem detetadas. No caso das imagens pré-operatórias, há sete classes correspondentes aos diferentes números de cabelos em cada folículo capilar, enquanto que nas imagens pós-operatórias, há duas classes correspondentes aos enxertos transplantados e aos vestígios de cicatriz.

Entre as redes neuronais existentes de detecção de objetos - com base em deep learning -, citando como exemplos, *RetinaNet*[15], *M2Det*[16], *YOLO V4*[17] e *EfficientDet*[18], o primeiro é o preferido para o sistema proposto. O *RetinaNet* é uma rede de detecção de objetos que utiliza a arquitetura *ResNet-101* como base, sendo capaz de detetar folículos, enxertos transplantados e vestígios de cicatrizes.

### 3.3.1.2 Cálculo da espessura do folículo

A abordagem utilizada para calcular a espessura do cabelo passa por dois estágios: segmentação baseada em *deep learning* e um novo método de cálculo de espessura. Existem várias redes de segmentação de imagens, como *SegNet* [19], *UNet* [20], *ERFNet* [21]; todos eles se apresentam como métodos de segmentação baseados em deep learning. A *SegNet* é preferida como método de segmentação, pois destaca-se pelo seu desempenho e velocidade. Após a segmentação, um novo algoritmo é desenvolvido para calcular a espessura do cabelo em escala micrométrica. Esse algoritmo utiliza a transformada de distância[22] para obter os valores de distância dos pixels e para calcular a espessura média dos fios de cabelo. O processo é repetido para cada folículo capilar, e um histograma de espessura é criado para cada área, eliminando valores anómalos no processo.

### 3.3.1.3 Análise Métrica

Os folículos capilares detetados, a espessura do cabelo calculada e o tamanho da área medida são integrados e usados no cálculo de desempenho na última etapa. No sistema *KEBOT*, são utilizados o  $FU/cm^2$  (unidades foliculares por  $cm^2$ ), o diâmetro (espessura do cabelo) e a Densidade Calculada para o cálculo de desempenho. O  $FU/cm^2$  representa o número total de unidades foliculares numa área de  $1cm^2$ . O diâmetro é a espessura média do cabelo do paciente. A Densidade Calculada é a média de fios de cabelo em um folículo.

Utilizando essas três métricas, o CV, que é um critério geral de avaliação de desempenho, é calculado usando a equação fornecida em (3.1):

$$CV = \left( \frac{FU}{cm^2} \right) \times \text{Calc.Dens.} \times \text{Diameter} \quad (3.1)$$

O CV fornece informações cruciais sobre a qualidade da área doadora, a contagem de enxertos necessários para cobertura mínima e a capacidade total doadora para um paciente. Para calcular corretamente o CV, são realizadas as seguintes etapas:

- Cálculo do  $FU/cm^2$ : A área é calculada e o número total de folículos detetados nessa

área é determinado. Esses valores são utilizados para calcular o  $FU/cm^2$  para cada área específica.

- Cálculo da Densidade Calculada: A partir dos mesmos folículos utilizados no passo anterior, a contagem total de fios de cabelo é dividida pelo número de folículos para obter a Densidade Calculada.
- Cálculo do Diâmetro: A espessura do cabelo é calculada apenas para os fios de cabelo na área selecionada. Esse valor é utilizado como Diâmetro.

Após o cálculo desses três parâmetros, o CV é calculado para cada área analisada, gerando um relatório individual para cada paciente.

## **3.4 Métodos tradicionais de VPC**

### **3.4.1 Segmentação de Unidades Foliculares**

A segmentação de imagem desempenha um papel crucial na classificação de folículos capilares, permitindo a identificação e a análise específica da região de interesse numa imagem. A segmentação é utilizada para separar a região do folículo capilar das demais áreas da imagem, facilitando a extração de características relevantes para a sua classificação.

Ao segmentar uma imagem de folículos capilares, é possível isolar a região de interesse por meio de técnicas de binarização e complementação da imagem. Além disso, é possível classificar os folículos de acordo com o número de cabelos presentes, como folículos com um, dois ou três cabelos.

Para a segmentação da imagem do folículo capilar, é possível detetar os pixels de contorno da região do folículo por meio do uso de filtros de contorno. Esses filtros permitem identificar os limites da região segmentada, que podem ser representados por vetores de contorno com comprimento e ângulo definidos. Esses vetores de contorno são representados por coordenadas de linha e coluna dos pontos do limite do folículo. A deteção dos vetores de contorno é realizada a partir do ponto de pixel do objeto mais à esquerda e superior, continuando no sentido horário dos ponteiros do relógio.

A segmentação de imagem é essencial para a classificação precisa e detalhada dos folículos capilares. Ao isolar a região de interesse e extrair informações relevantes dos limites do folículo, é possível obter características distintas que facilitam a classificação de acordo com o número de cabelos presentes. Essa abordagem permite um melhor entendimento da estrutura e distribuição dos folículos capilares, auxiliando em estudos dermatológicos, pesquisas sobre perda de cabelo e desenvolvimento de tratamentos específicos para cada tipo de folículo capilar. [11]

## 3.5 Redes Neurais

### 3.5.1 YOLO: "You Only Look Once"

*YOLO* ("You Only Look Once") é um dos algoritmos mais inovadores e eficientes na área de detecção de objetos em imagens. Este, apresenta uma abordagem única que oferece vantagens significativas em relação a outros métodos de detecção existentes.

Uma das suas principais características é a sua capacidade de processar imagens em tempo real. Enquanto outros algoritmos baseados em janelas deslizantes exigem múltiplas passagens pela imagem, o *YOLO* realiza a detecção em uma única etapa. Isso significa que, em vez de avaliar regiões numa imagem, o *YOLO* analisa logo a imagem completa, identificando e localizando os objetos com apenas uma inferência. Essa abordagem torna o *YOLO* extremamente rápido e adequado para aplicações em tempo real.

Outra grande vantagem do *YOLO* é a sua capacidade de detetar objetos de diferentes tamanhos. Ao dividir a imagem em células e atribuir responsabilidades a cada célula, o *YOLO* é capaz de identificar objetos pequenos, como pedestres, assim como objetos maiores, como carros. Isto torna a rede mais versátil e adequada para diferentes cenários de detecção. [23, 24]

Comparado com outros métodos de detecção, destaca-se em termos de desempenho em tempo real e facilidade de implementação. Enquanto os métodos tradicionais exigem várias etapas de processamento e computacionalmente intensivas, o *YOLO* é eficiente e pode ser executado em tempo real, mesmo em dispositivos com recursos limitados. Além disso, o *YOLO* oferece uma abordagem "end-to-end", o que significa que todo o processo de detecção é realizado numa única rede neuronal, permitindo uma otimização direta do desempenho da detecção.

No entanto, é importante notar que o *YOLO* pode apresentar algumas limitações em relação à precisão em comparação com métodos mais complexos.[24]

A utilização da rede *YOLO* no contexto de detecção de unidades foliculares em tempo real é uma solução bastante viável, devido à sua rapidez de detecção, pelo facto de ser preciso analisar a imagem uma só vez e também a sua capacidade de detetar múltiplos objetos numa única imagem.

### 3.5.2 R-CNN: "Regions with CNN feature & Fast/Faster-RCN"

*R-CNN* ("Regions with CNN feature") é uma abordagem pioneira no campo da detecção de objetos, que revolucionou a forma como os objetos são localizados e classificados em imagens.[25] Métodos tradicionais de detecção de objetos contavam com janelas deslizantes e características manualmente projetadas para detetar objetos. No entanto, *R-CNN* introduziu um conceito inovador de propostas de região combinadas com redes neurais convolucionais (CNN) para alcançar resultados de ponta na detecção de objetos.

A ideia principal por trás da *R-CNN* é gerar propostas de região numa imagem, que são possíveis localizações de caixas delimitadoras contendo objetos. Essas propostas de região são obtidas usando o método de procura seletiva ("selective search") [26], um algoritmo popular

que gera de forma eficiente um conjunto diversificado de regiões candidatas com base em segmentação de imagem e medidas de similaridade. Em vez de examinar exaustivamente toda a imagem, o R-CNN concentra-se apenas no processamento dessas propostas de região, reduzindo significativamente a complexidade computacional.

Uma vez que as propostas de região são geradas, o R-CNN extrai características da CNN de cada região. Essas características são obtidas ao passar as propostas de região por uma CNN pré-treinada, como ImageNet[27] ou VGGNet[28], que foi inicialmente treinada para classificação de imagens. A CNN extrai representações visuais de alto nível das propostas de região, permitindo uma melhor compreensão dos objetos presentes em cada região.

Uma das vantagens do R-CNN é a sua capacidade de aprender características discriminativas diretamente dos dados. Ao utilizar CNNs, o R-CNN pode capturar informações visuais ricas das propostas de região, levando a um melhor desempenho na detecção de objetos. Além disso, o R-CNN beneficia da disponibilidade de conjuntos de dados rotulados em grande escala, permitindo o pré-treino de CNNs em grandes coleções de imagens, o que fornece representações visuais genéricas valiosas.

No entanto, a principal desvantagem é a sua lentidão na inferência devido ao processamento sequencial de cada proposta de região, o que o torna impraticável para aplicações em tempo real. Para resolver este problema, trabalhos subsequentes, como o *Fast R-CNN*[29] e o *Faster R-CNN*[30], foram introduzidos, melhorando a eficiência do R-CNN integrando a geração de propostas de região e extração de características da CNN numa única rede neuronal.

O *Fast R-CNN* melhorou o *R-CNN* ao substituir a etapa de extração de características por uma camada de rede neuronal convolucional compartilhada. Em vez de extrair as características para cada proposta de região separadamente, o *Fast R-CNN* realiza a extração de características uma única vez para a imagem completa. Em seguida, essas características são usadas para selecionar as regiões de interesse (ROIs) correspondentes às propostas de região. As ROIs são posteriormente redimensionadas para um tamanho fixo e alimentadas numa camada totalmente conectada, que realiza tanto a classificação quanto a regressão das caixas delimitadoras.

Já o *Faster R-CNN* trouxe um aprimoramento adicional ao *Fast R-CNN* ao introduzir a arquitetura de rede neuronal chamada de "*Region Proposal Network*" (*RPN*). O *RPN* é responsável pela geração eficiente das propostas de região diretamente da rede neuronal convolucional. Em vez de usar o método de procura seletiva para gerar as propostas de região, o *RPN* utiliza uma rede neuronal para aprender a prever as regiões de interesse diretamente da imagem. Essa abordagem elimina a necessidade de uma etapa separada de geração de propostas de região, tornando o processo mais rápido e eficiente.

O *Fast R-CNN* e o *Faster R-CNN* oferecem uma detecção mais precisa, mas podem ser mais lentos em termos de tempo de processamento. Sendo o foco deste trabalho, a detecção de unidades foliculares, é preciso conjugar com as necessidades que forem apresentadas para escolher a melhor rede para desempenhar o papel de detecção.



# 4

## Trabalho Desenvolvido

Neste ponto do trabalho, apresentarei uma visão geral do trabalho que foi desenvolvido para a criação deste projeto, que se concentra na detecção orientada e no tracking de unidades foliculares.

Para atingir esse objetivo, foram realizadas diversas etapas e processos. Inicialmente, foi realizada uma pesquisa abrangente sobre transplante capilar, as características das unidades foliculares e as técnicas existentes de detecção e tracking. Essa pesquisa permitiu compreender o estado atual do conhecimento e identificar áreas que poderiam ser aprimoradas.

### 4.1 Data

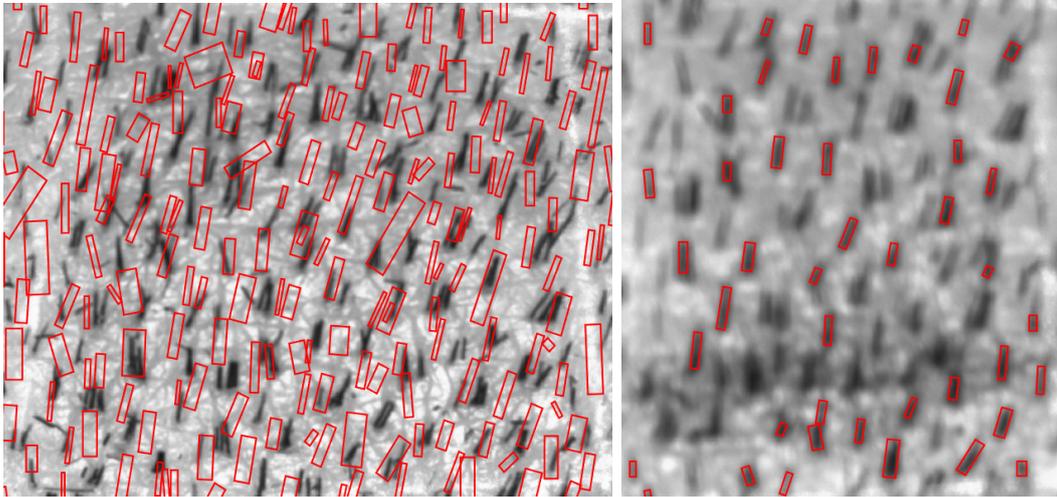
#### 4.1.1 Verificação de Imagem

Em relação a todo o conjunto de dados de treino existente, é importante ressaltar que as imagens utilizadas neste trabalho foram capturadas por meio do software desenvolvido pela empresa Insparya. Esse software realiza a detecção de unidades foliculares e fornece uma *bounding box* ao redor de cada uma delas. A obtenção dessas imagens foi realizada utilizando *bags* em ROS, que são pacotes de dados desenvolvidos pela própria empresa. Através do ajuste dos parâmetros de configuração, como *threshold* de detecção máximo e mínimo, tamanho e largura dos folículos, margens de interseção, entre outros, foi possível obter múltiplas detecções de unidades foliculares numa única imagem, sendo cada detecção representada por uma *bounding box* orientada. Quanto mais detecções forem obtidas, melhor será a construção do *dataset* a ser utilizado em uma rede neuronal para um posterior treino, para ser utilizado como meio principal de detecção.

É importante ressaltar que cada *bounding box* fornece informações precisas sobre as coordenadas (x, y) dos quatro cantos que a compõem, totalizando oito valores. Esses valores são armazenados num arquivo de texto, onde cada linha representa uma *bounding box* e, conseqüentemente, a detecção de um folículo.

Com o objetivo de verificar a qualidade das imagens e das detecções realizadas pelo software, foi desenvolvido um código em MATLAB que desenha todas as *bounding boxes* dos folículos detetados numa imagem, para que seja possível fazer uma análise da imagem em si e também avaliar a qualidade das suas respectivas detecções, com o intuito de descartar imagens excessivamente desfocadas e aquelas que tenham muito poucas detecções, que não sejam adequadas para a inclusão no *dataset* a ser criado.

Por meio desse processo de verificação de qualidade, é possível garantir que apenas imagens desejadas sejam utilizadas. Além disso, é possível identificar possíveis deficiências nas detecções realizadas, possibilitando melhorias e ajustes de forma a obter resultados mais precisos e rigorosos.



**Figure 4.1:** Imagens Mal Anotadas/Desfocadas.

A Figura 4.1 representa o tipo de imagens que era necessário remover do conjunto de dados. São imagens fornecidas pela empresa as quais vinham mal anotadas ou não eram de interesse a sua inclusão no conjunto de dados de treino.

#### 4.1.2 Conversão de Formato das *labels*

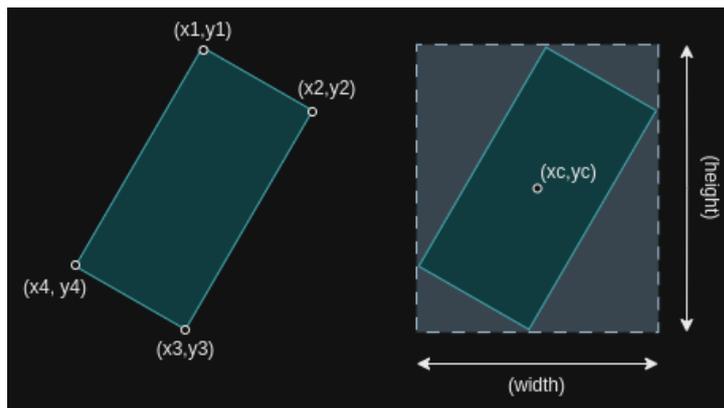
Como mencionado anteriormente, o formato fornecido pelo software da empresa para as coordenadas de *bounding box* é:

$$\langle x1, y1, x2, y2, x3, y3, x4, y4 \rangle$$

Cada um destes conjuntos de valores corresponde às coordenadas de um canto da *bounding box* que delimita uma unidade folicular. No entanto, para utilizar essas coordenadas de *bounding box* na rede *YOLO*, é necessário convertê-las para um formato específico. Para criar uma *bounding box* no formato correto para a entrada da rede Yolo, é necessário organizar as *labels* na seguinte configuração:

$$\langle \text{object} - \text{class} \rangle, \langle xc \rangle, \langle yc \rangle, \langle width \rangle, \langle height \rangle$$

Esta conversão é feita tal como mostrado na figura 4.2.



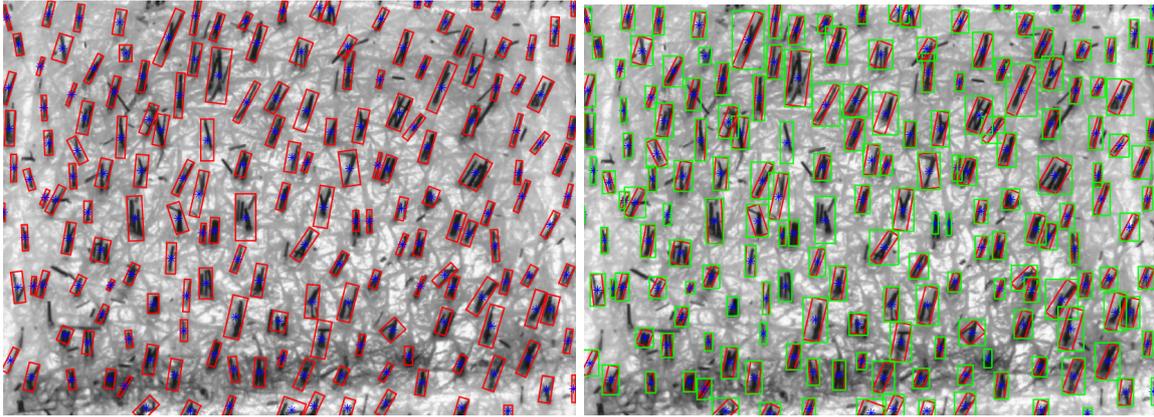
**Figure 4.2:** Transformação formato de *labels*.

Cada um destes valores tem o seguinte significado:

- $\langle \text{object class} \rangle$  Este é um valor inteiro que corresponde ao ID de cada classe, como a única classe é a unidade folicular, será apenas utilizado o valor 0. O valor das classes pode variar entre  $\langle 0 \rangle$  e o  $\langle \text{número de classes} - 1 \rangle$ .
- $\langle xc \rangle$  Representa a coordenada x do centro do retângulo delimitador. Esta coordenada é calculada encontrando os valores máximos e mínimos de x e, em seguida, obtendo o valor absoluto dessa diferença. Esse valor é dividido pelo tamanho da imagem, resultando no valor central de x.
- $\langle yc \rangle$  Esta coordenada é obtida da mesma maneira que o valor de "xc", porém, em vez de utilizar o eixo de coordenadas "x", são usados os valores do eixo de coordenadas "y".
- $\langle \text{width} \rangle$  É a diferença do valor de x máximo com o valor de x mínimo.
- $\langle \text{height} \rangle$  É a diferença do valor de y máximo com o valor de y mínimo.

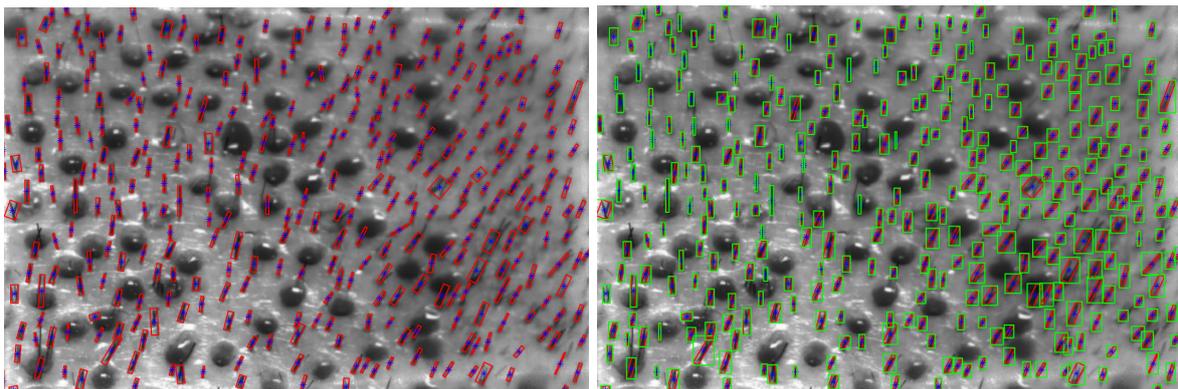
Nas Figura 4.3, é possível observar a representação visual dessa transformação em duas etapas. Inicialmente, é apresentada a *bounding box* orientada, seguida da adição da *bounding box* com o formato característico do modelo *YOLOv5*. Esta análise tem como objetivo a comparação da adequação das novas *bounding boxes* na delimitação dos folículos.

Ao realizar essa comparação, é feita a avaliação como os folículos estão delimitados pelas diferentes formas de *bounding boxes*, permitindo uma análise mais precisa da eficácia da transformação.



**Figure 4.3:** Transformação formato de *labels* (Exemplo 1).

A ilustração representada na Figura 4.4 retrata a aplicação da transformação mencionada anteriormente, porém, desta vez, numa imagem consideravelmente mais complexa. Nesta imagem, os folículos capilares encontram-se numa escala bastante reduzida, o que torna mais difícil a sua deteção. Além disso, surge uma complexidade adicional: tornam-se visíveis as extrações que foram efetuadas por meio dos orifícios presentes na região craniana do paciente.



**Figure 4.4:** Transformação formato de *labels*. (Exemplo 2)

Foi também realizado um procedimento de adaptação de *labels* do formato fornecido pela empresa para o formato da rede *YOLOv5\_OBB*, visando a geração de um conjunto de dados, o maior possível, a partir dos dados previamente existentes.

Para a rede *YOLOv5\_OBB*, as *labels* de entrada têm de estar na seguinte configuração:

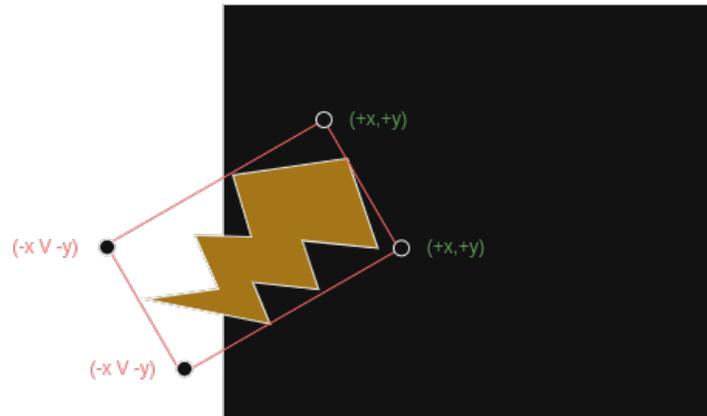
$$x1, y1, x2, y2, x3, y3, x4, y4, < class - name >, < difficulty >$$

Nesse sentido, foi requerida uma simples conversão das coordenadas de cada ponto para as dimensões reais da imagem que seria utilizada na rede. Isto deve-se ao fato de o software da empresa fornecer valores de *x* e *y* na escala entre 0 e 1. Além disso, houve a necessidade de incluir o nome da classe, denominada de "*uf*", que corresponde à unidade folicular, sendo a única classe a ser utilizada.

A dificuldade é um parâmetro presente a ser considerado, podendo assumir os valores 0 ou 1.

No entanto, dado que utilizamos apenas uma classe, todas as entradas foram definidas como 0. Nesse sentido, a dificuldade é uniforme, visto que a classe é única em todas as situações.

É relevante destacar que o software da empresa gerava *bounding boxes* que extrapolavam os limites da imagem, resultando em valores negativos. Isto causava a criação de imagens corrompidas para o treino das redes em questão. A fim de mitigar este problema, sempre que um valor negativo era detectado, ele era convertido para 0, que é o valor mais próximo dentro dos limites da imagem.



**Figure 4.5:** Extrapolação da Imagem

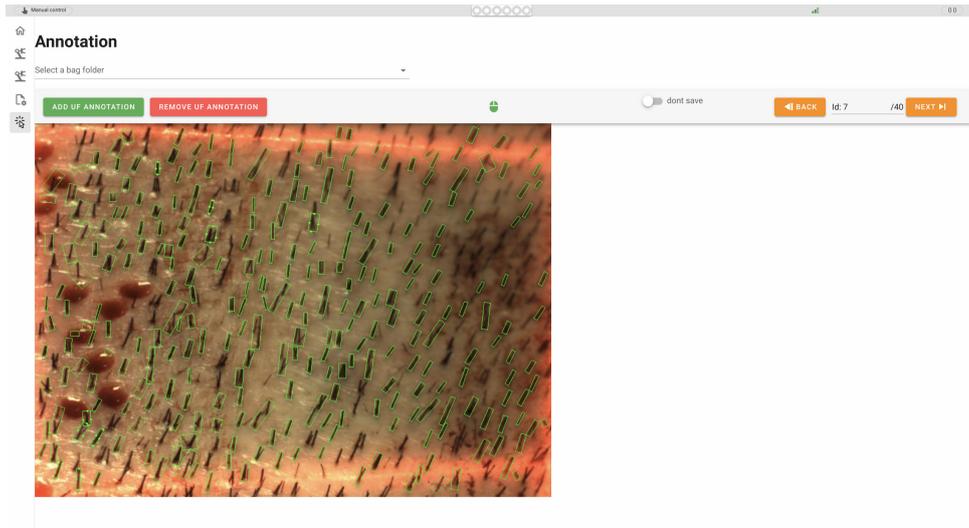
A Figura 4.5 exemplifica a situação mencionada anteriormente. Neste caso, se os valores negativos de  $x$  e/ou  $y$  se estenderem além da imagem, como indicado pelos pontos a vermelho, é necessário ajustá-los para o ponto mais próximo dentro da imagem, que é o ponto de coordenadas 0.

### 4.1.3 Anotações de Imagens

A empresa dispõe de um software exclusivamente desenvolvido para a marcação de anotações de unidades foliculares em imagens, o qual é utilizado com o objetivo de maximizar a geração de imagens dentro de um período de tempo limitado. A disponibilidade reduzida dos enfermeiros para a anotação de imagens destinadas ao conjunto de dados impulsionou a necessidade de encontrar uma solução alternativa para prosseguir com as anotações e assim foi possível a anotação de imagens para complementar as imagens anotadas pela empresa. Considerando que cada imagem contém, em média, cerca de 200 unidades foliculares, a tarefa de anotar uma única imagem é notavelmente demorada.

O programa opera da seguinte maneira:

Primeiramente pela definição de 2 pontos para determinar a orientação da linha dos folículos e seguidos de outros 2 pontos para indicar sua espessura, o software gera automaticamente uma *bounding box* em torno da unidade folicular em questão. No entanto, é necessário repetir esse procedimento para cada um dos múltiplos folículos presentes na imagem. Esse processo, apesar de minucioso, é fundamental para a precisão e eficácia das análises subsequentes.



**Figure 4.6:** Programa de Anotações Desenvolvido pela Empresa

A Figura 4.6 representa a interface do programa utilizado, e uma imagem com quase todas as anotações das suas unidades foliculares. Este processo foi repetido para a maior variedade de imagens, de modo a encontrar o *dataset* mais variado possível. Com a ajuda dos enfermeiros, foi possível a criação de um conjunto de dados consideravelmente imprescindível para o treino das redes em questão.

## 4.2 Cálculo das métricas

Medir o desempenho dos modelos e algoritmos utilizados é fundamental para avaliar a qualidade das soluções propostas e para a tomada de decisões com base na avaliação obtida. As métricas de avaliação desempenham um papel crucial neste processo, permitindo que os profissionais compreendam o quão bem os seus modelos funcionam em relação aos objetivos estabelecidos.

Entre as métricas mais utilizadas, estão a *Precision*, *Recall*, *Accuracy* e o *F1-Score*. Cada uma delas aborda diferentes aspetos do desempenho do modelo e fornece a informação necessária sobre o seu comportamento em diferentes cenários.

A *Precision* é uma métrica essencial que avalia a proporção de exemplos positivos corretamente classificados em relação a todos os exemplos classificados como positivos pelo modelo. Noutras palavras, mede a capacidade do modelo de evitar falsos positivos. A precisão é especialmente relevante em situações em que os falsos positivos têm consequências significativas, como em diagnósticos médicos. Portanto, uma alta precisão indica que o modelo é confiável ao rotular exemplos como positivos.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

O *Recall*, avalia a capacidade do modelo de capturar todos os exemplos relevantes da classe positiva, minimizando falsos negativos. Esta métrica é crucial em cenários que é necessário evitar falsos negativos, como na deteção de doenças. Um alto *Recall* indica que o modelo é eficaz em identificar a maioria dos exemplos positivos.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

A *Accuracy* é uma métrica amplamente utilizada que mede a proporção de exemplos corretamente classificados em relação ao total de exemplos. Embora seja uma métrica de fácil compreensão, pode ser enganosa em cenários onde as classes não estão balanceadas. Por exemplo, num conjunto de dados onde uma classe é dominante, um alto nível de *Accuracy* pode ser alcançado simplesmente classificando todos os exemplos como pertencentes à classe dominante. Portanto, a *Accuracy* deve ser interpretada com cuidado e considerada juntamente com outras métricas. Isto não será um problema devido à existência de apenas uma única classe.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FN} + \text{FP} + \text{TN}}$$

O *F1-Score* é uma métrica que combina a *Precision* e o *Recall* numa única medida. Este calcula a média harmónica entre estas duas métricas, fornecendo uma maneira de avaliar o equilíbrio entre a capacidade do modelo de evitar falsos positivos e falsos negativos. O *F1-Score* é particularmente útil quando há um trade-off entre *Precision* e *Recall*, ajudando a encontrar um ponto ideal de equilíbrio entre as duas métricas.

$$\text{F1-Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

## 4.3 Redes Neurais Sem Orientação

### 4.3.1 YOLOv5

A rede *YOLOv5* surge como uma evolução significativa na abordagem de deteção em tempo real. O acrónimo "YOLO" significa "You Only Look Once", expressando a ideia central por trás desse algoritmo: realizar a deteção numa única passagem pela rede neuronal, maximizando a eficiência e minimizando o tempo de processamento.

A arquitetura do *YOLOv5* é composta por três componentes-chave:

- *Backbone*: Este é o núcleo central da rede. No *YOLOv5*, o *Backbone* é construído usando a estrutura *New CSP-Darknet53*, uma modificação da arquitetura *Darknet* utilizada em versões anteriores.

- *Neck*: Essa parte atua como uma ponte entre o *Backbone* e a *Head*. No *YOLOv5*, utilizam-se as estruturas *SPPF* e *New CSP-PAN*.
- *Head*: A Head é responsável por gerar a saída final. O *YOLOv5* utiliza a Cabeça do *YOLOv3* para essa finalidade.

Na Figura 4.7, é possível observar a estrutura da rede *YOLOv5*, que utiliza uma abordagem de detecção de múltiplas escalas, dividindo a imagem de entrada em diferentes grelhas e executando detecções em cada uma delas. Esse método permite que o algoritmo identifique objetos de diversos tamanhos de maneira eficaz. [4]

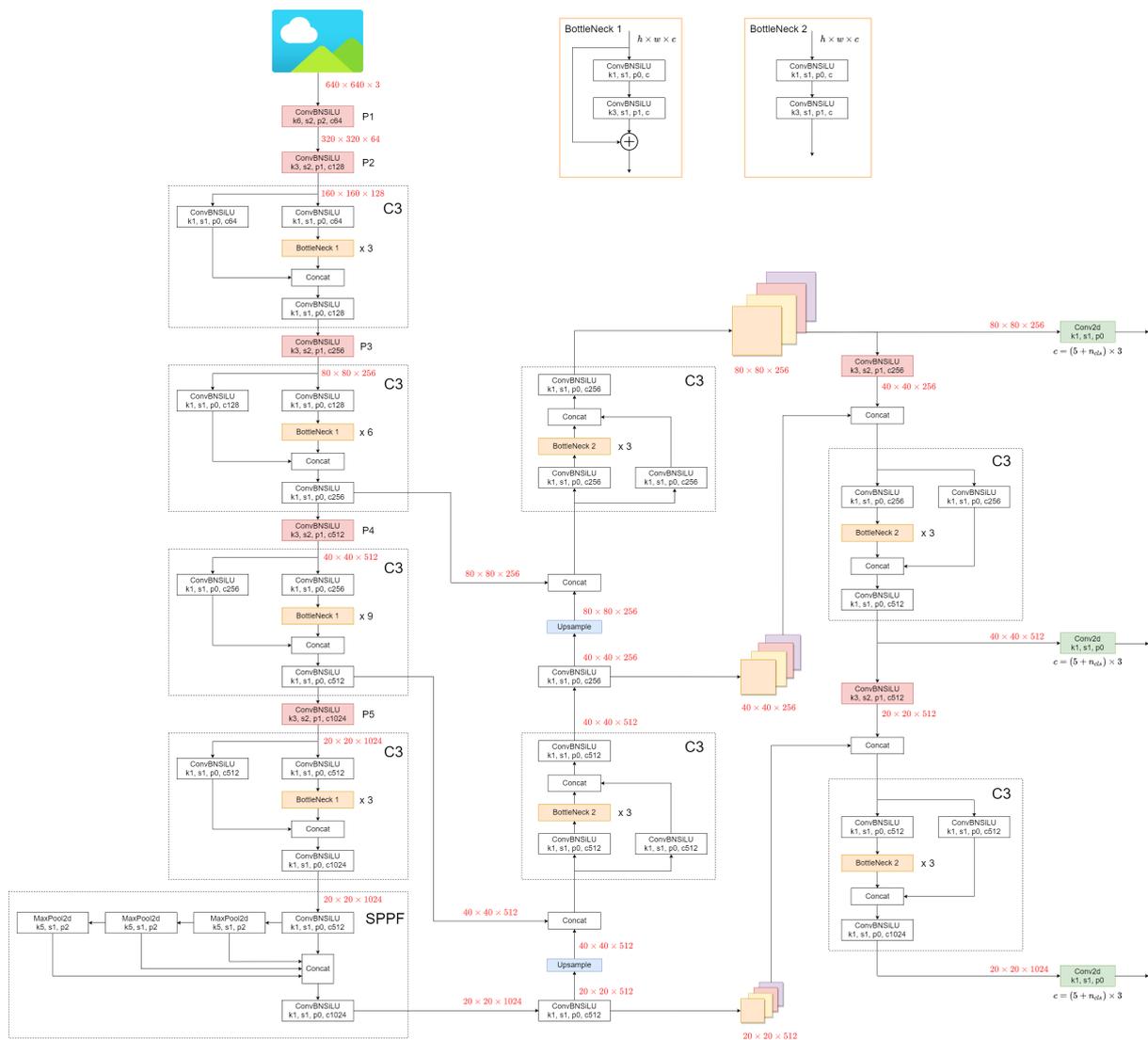


Figure 4.7: Estrutura do Modelo *YOLOv5* [4]

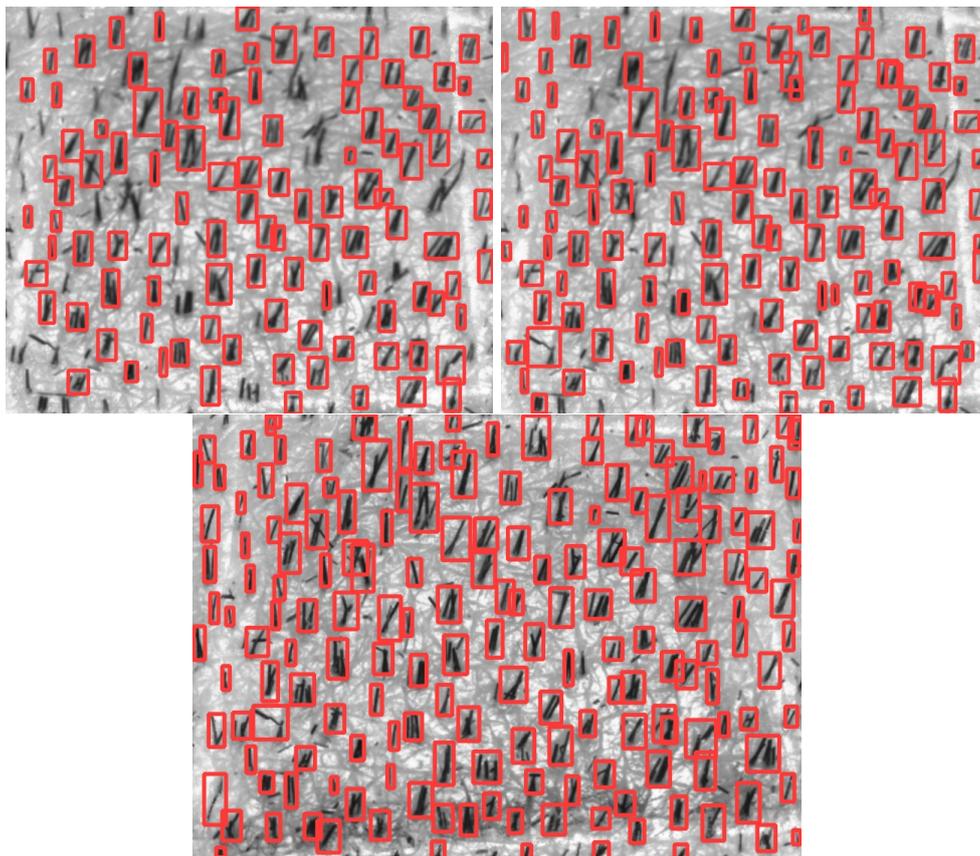
Esta foi uma rede escolhida pela sua popularidade, rapidez e de fácil utilização, sendo que é apenas de teste para o propósito final, pois o principal objetivo, é uma rede que faz imediatamente a detecção orientada, devolvendo como output a pose do folículo. Sendo esta rede sem orientação, ao receber os seus dados de detecção, calcular posteriormente a sua orientação dentro de cada *bounding box* iria exigir um grande custo computacional e também um tempo

de computação total muito maior, que é exatamente o que se pretende diminuir.

Mas como ponto de partida, a escolha inicial desta rede excedeu as expectativas pois os resultados obtidos através da mesma, foram bastante promissores, como é explicado mais à frente.

Com isto, foi feito um treino com o *dataset* já existente, que apesar de pequeno, serviu para testar a rede *YOLOv5* no ambiente a ser utilizada, na detecção de unidades foliculares. Foi usado um conjunto de imagens, todas a preto e branco, nas quais estavam todos os folículos de uma imagem anotados. O *dataset* continha 147 imagens etiquetadas para a parte do treino e 33 para a validação, isto para fazer uma divisão por volta dos 70% para treino e 30% para validação.

Na Figura 4.8 está apresentada a evolução das detecções efetuadas pela rede *YOLOv5*, cada uma derivada de um treino diferente. Nas duas primeiras representações foi utilizada a mesma imagem, para existir um termo de comparação e verificar se houve evolução. Na terceira representação foi utilizada a mesma imagem, mas ligeiramente deslocada para observar como se comporta num cenário minimamente diferente.

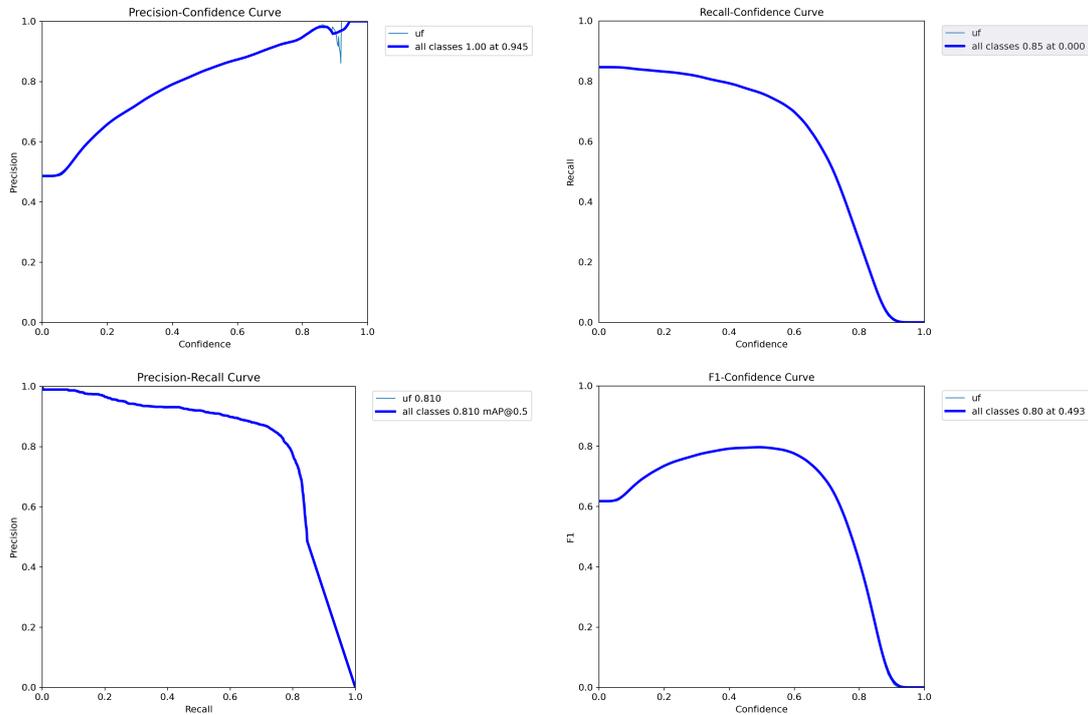


**Figure 4.8:** Evolução da Detecção de Unidades Foliculares

A partir das imagens obtidas observa-se um crescimento significativo na qualidade dos resultados obtidos, sendo a última imagem o resultado do treino que será exposto a seguir, com os seguintes parâmetros de treino, utilizando o pré-treino com o *dataset* "small" da rede *YOLOv5*, "yolov5s.pt":

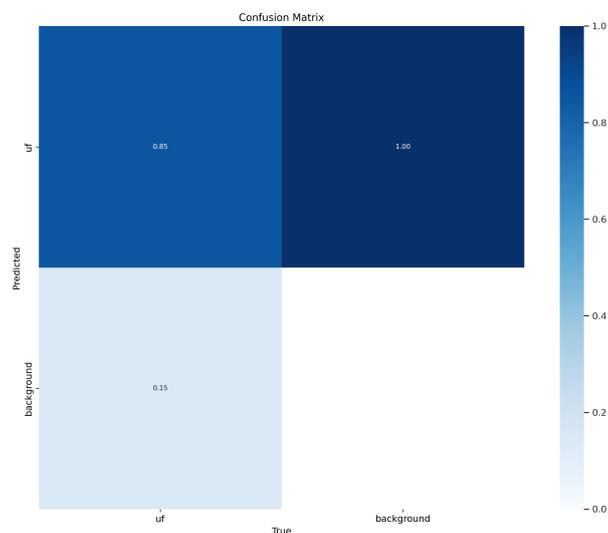
epochs=100, imgsz=640, batch=16, weights=yolov5s.pt

Na Figura 4.9 observam-se os gráficos resultantes do treino anteriormente mencionado, tais como *Precision*, *Recall*, *Precision-Recall* e *F1Score*, respetivamente. É possível ver o seu comportamento consoante o grau de confiança utilizado na deteção das unidades foliculares.



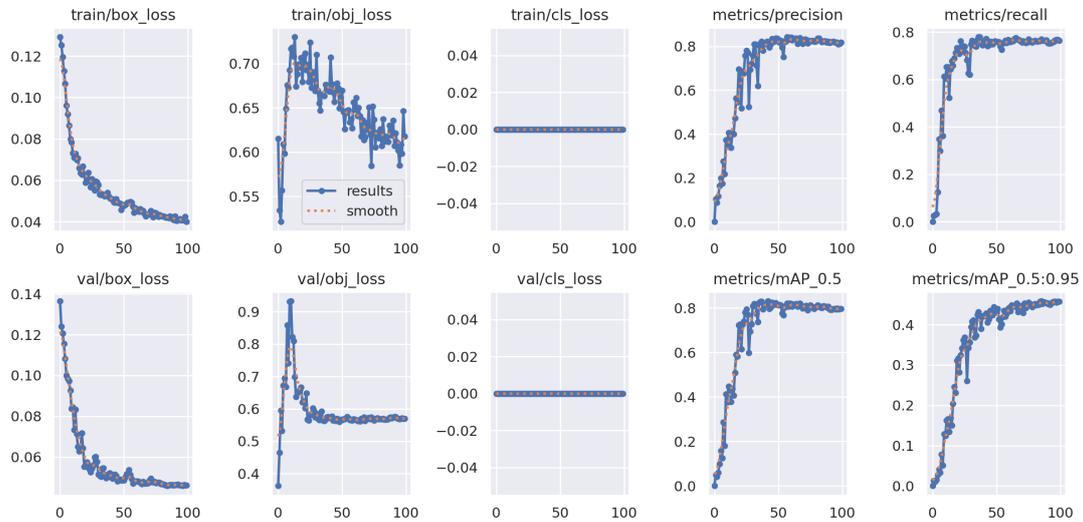
**Figure 4.9:** Métricas do Treino da Rede YOLOv5

A Figura 4.10 exibe a matriz de confusão, ilustrando as percentagens de *True Positives (TP)*, *False Positives (FP)*, *True Negatives (TN)* e *False Negatives (FN)*. A partir desses valores, torna-se possível o cálculo das métricas relevantes para uma posterior análise.



**Figure 4.10:** Matriz de Confusão da Rede YOLOv5

Os gráficos de perdas e das restantes métricas, tais como *mAP*, *Precision* e *Recall* são apresentados na Figura 4.11. No gráfico de perdas, foram obtidos valores significativamente aceitáveis, mas ainda podem vir a ser melhorados, ou seja diminuídos.



**Figure 4.11:** Resultados da Rede YOLOv5

Relativamente aos valores de *Precision*, *Recall* e *mean Average Precision* foram obtidos os seguintes valores ao fazer a validação derivada do treino da rede:

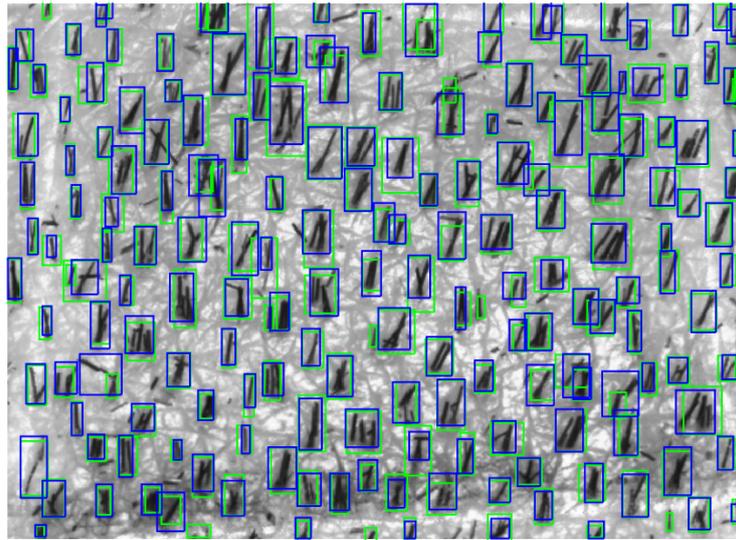
**Table 4.1:** Métricas Derivadas do Treino da Rede YOLOv5

	Valor(0-1)	Percentagem(0-100)
<b>Precision</b>	0.823	82.3%
<b>Recall</b>	0.769	76.9%
<b>mAP50</b>	0.809	80.9%
<b>mAP50-95</b>	0.458	45.8%

Estes valores indicam um desempenho razoavelmente bom do modelo de rede neuronal em termos de deteção e classificação, mas pode haver margem para melhoria, especialmente na redução de falsos negativos e na expansão da faixa de confiança para uma deteção mais precisa em diferentes situações.

Estes resultados refletem a contínua otimização alcançada através do treino com o conjunto de dados limitado disponível naquele momento. Com o acréscimo de dados adicionais devidamente anotados e a inclusão de uma maior diversidade de cenários possíveis, é de esperar um progresso ainda maior, superando os ganhos já alcançados até então.

Estes valores, resultaram então na seguinte deteção visível na Figura 4.12, a qual já mostra a comparação do comportamento da rede YOLOv5 com o nosso "ground truth", que são as imagens etiquetadas pelos enfermeiros que trabalham na empresa Insparya.



**Figure 4.12:** Comparação de unidades foliculares detetadas pela rede *YOLO* (Azul) e pelo software da empresa Insparya (Verde)

É possível observar na Figura 4.12, as *bounding boxes* a verde, que representam as *labels* anotadas pelos enfermeiros e a azul as deteções efetuadas através da rede *YOLOv5*, isto sem orientação. Pelos resultados obtidos, consegue-se perceber que o comportamento da rede foi positivo, com falhas mínimas na deteção. Quase todas as *bounding boxes* a verde, estão correspondidas com uma *bounding box* a azul.

### 4.3.2 *YOLOv8*

*YOLOv8* representa o modelo mais recente, estado-da-arte (SOTA), que se baseia nos êxitos das versões anteriores do *YOLO*(You Only Look Once), ao mesmo tempo introduz novas funcionalidades e aperfeiçoamentos para elevar ainda mais o seu desempenho e adaptabilidade. O *YOLOv8* foi concebido para ser rápido, preciso e de fácil utilização, sendo que esta versão para além das tarefas base, deteção de objetos e classificação de imagens, trouxe mais três novidades, sendo estas:

- *Segmentação* - A segmentação é um processo que consiste em particionar uma imagem em diversas áreas, levando em consideração o conteúdo presente. Cada área recebe uma etiqueta de acordo com o seu conteúdo.
- *Pose* - Trata-se de uma atividade que consiste em reconhecer pontos específicos numa imagem. Esses pontos são denominados de pontos-chave e têm a finalidade de rastrear movimentos ou realizar estimativas de pose.
- *Tracking*: O seguimento de objetos é uma atividade que requer a identificação da posição e da categoria dos objetos, seguida da atribuição de um ID exclusivo a essa deteção em sequências de vídeo.

Na Figura 4.13 está representada a estrutura detalhada da rede *YOLOv8*.

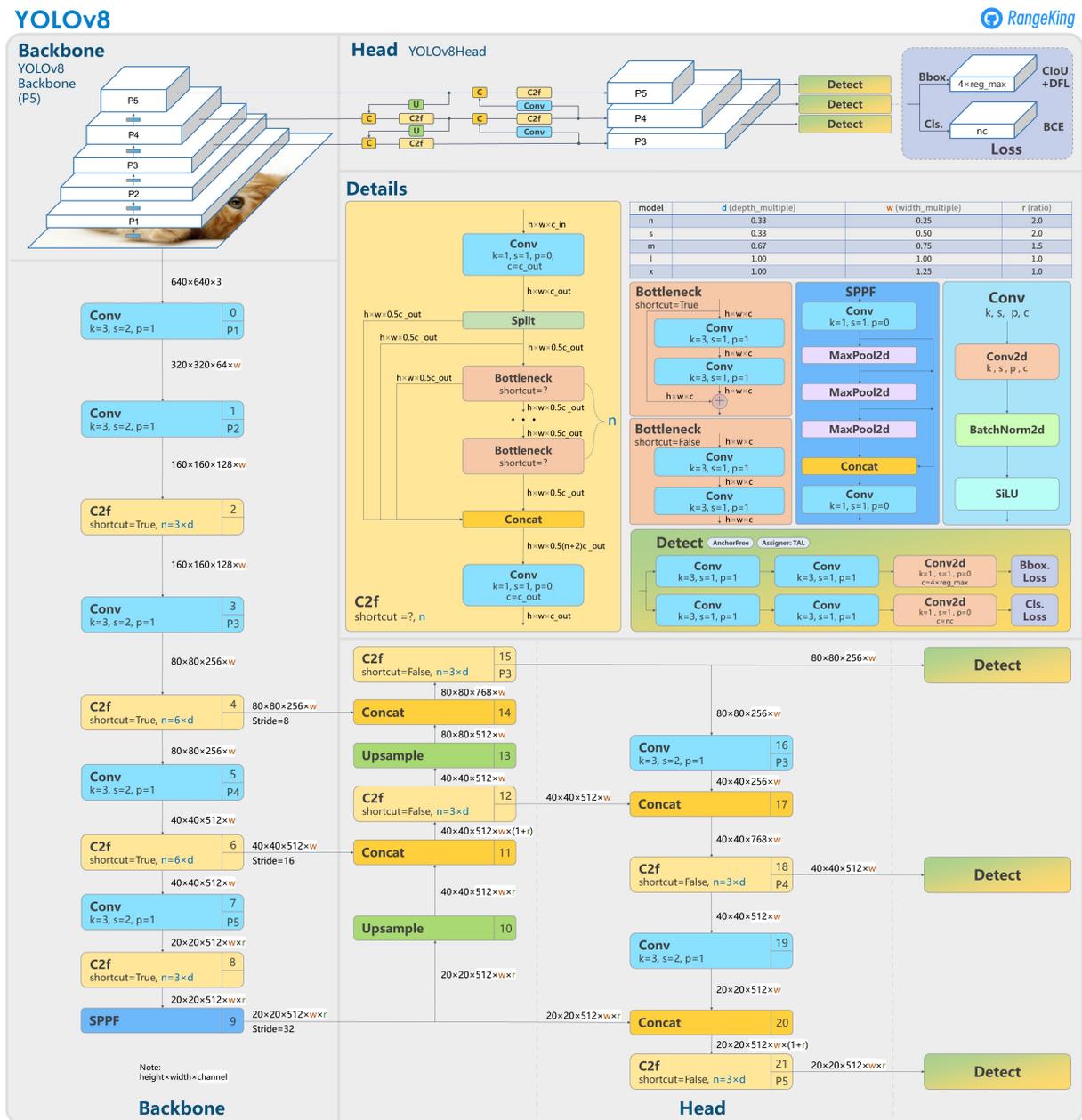
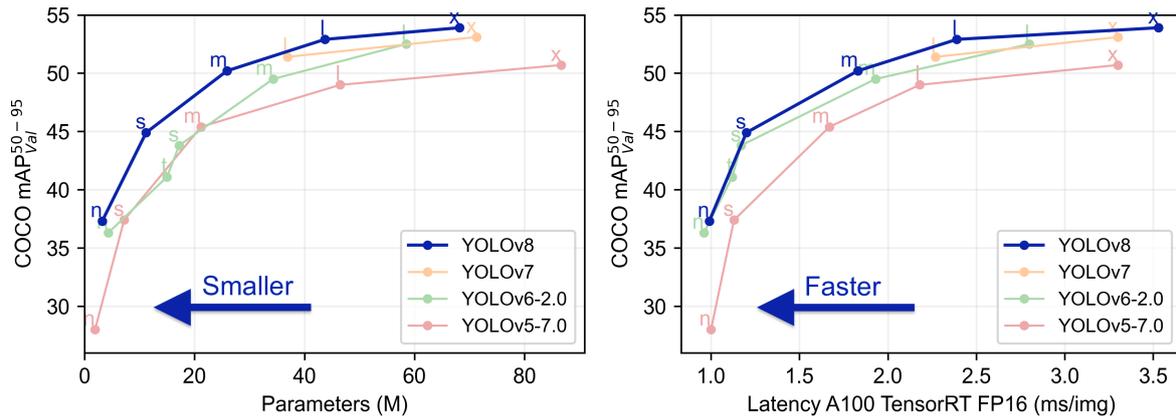


Figure 4.13: Estrutura da rede *YOLOv8* retirado de [5]

O diagrama ilustrado na Figura 4.14 retrata o comportamento das diferentes versões da rede *YOLO* relativamente ao *mAP* ("mean Average Precision") e *Latency* usando os diferentes conjuntos de dados, n (nano), s (small), m (medium), l (large) e x (extra large).



**Figure 4.14:** Comparação versões YOLO relativamente ao  $mAP$  e  $Latency$ [5]

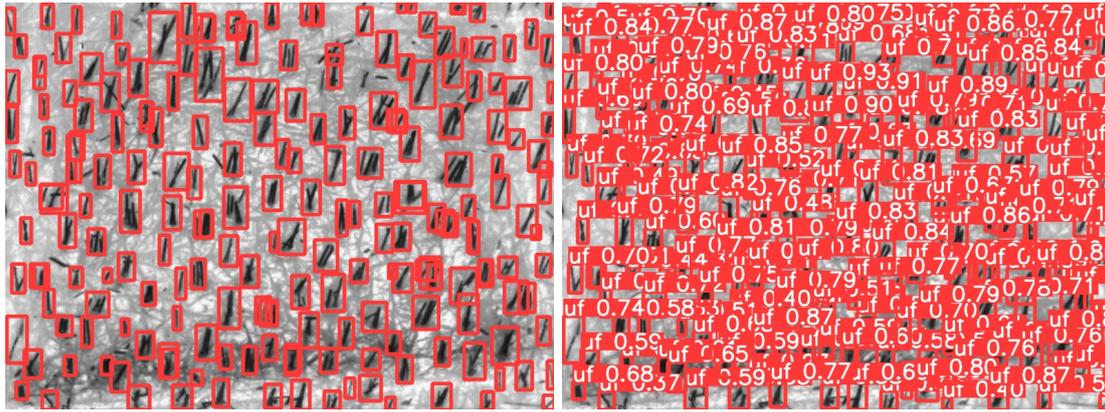
- $mAP$  (*mean Average Precision*) - é uma métrica usada para avaliar a precisão de algoritmos de deteção e classificação de objetos em tarefas de visão computacional, calculando a precisão média em diferentes níveis de confiança para várias classes de objetos.
- Latência - refere-se ao mínimo período de tempo de atraso ou espera em sistemas de processamento de dados ou comunicação, indicando um intervalo de tempo muito breve para os dados percorrerem entre pontos de entrada e saída.

A rede *YOLOv8* destaca-se, no sentido de equilibrar a sua grande precisão com uma elevada velocidade de inferência.

A configuração de *labels* desta rede é exatamente igual à rede *YOLOv5*. Portanto, dar início aos testes da rede foi consideravelmente simples, dado que já existia um conjunto de dados semi-preparado, que havia sido utilizado no treino da rede *YOLOv5*.

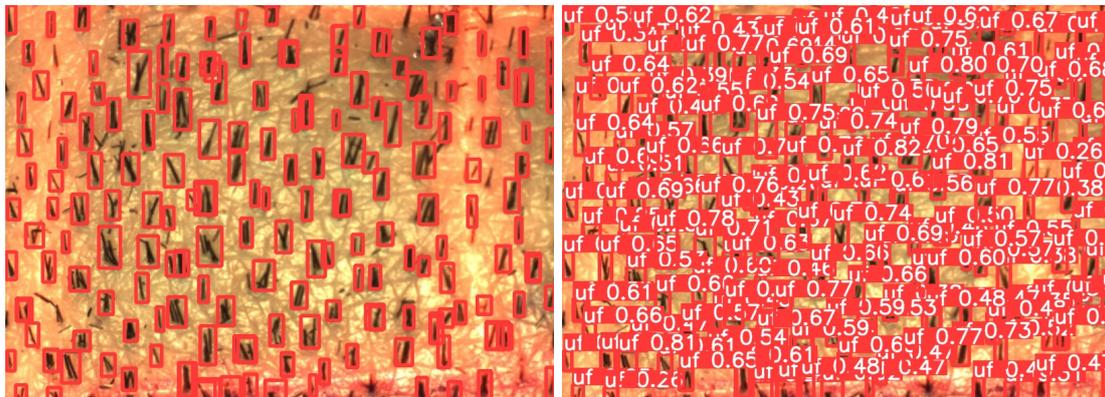
A rede foi testada com o melhor *dataset* de treino utilizado na rede *YOLOv5*, que apesar de ser um conjunto de dados reduzido e apenas possuir imagens a preto e branco apresentou resultados com um grau de "precision" e "recall" bastante elevado.

A Figura 4.15 apresenta uma deteção realizada utilizando a rede *YOLOv8*, a partir de uma imagem a preto e branco do couro cabeludo de um paciente. É notável que todos os folículos presentes na imagem detetada estão devidamente identificados, mesmo perante a presença das consideráveis variações de tamanho dos cabelos. A performance da rede foi satisfatória, considerando o tamanho relativamente reduzido do conjunto de dados de treino utilizado. Isto sugere que, aumentando o número de imagens para o treino da rede, seria possível alcançar níveis elevados de precisão nas deteções das unidades foliculares. Nas imagens destinadas à deteção, encontram-se apenas o fundo (pele do couro cabeludo) e as unidades foliculares em si. Conforme os folículos são extraídos, começam a surgir orifícios, acompanhados de pequenas quantidades de sangue decorrentes desses orifícios, o que introduz mais interferência visual na imagem. Contudo, para abordar essa questão, é essencial dispor de um conjunto de dados abrangente que englobe todas as variações e cenários possíveis, visando assegurar uma maior solidez e confiabilidade.



**Figure 4.15:** Detecções *YOLOv8* a Preto e Branco

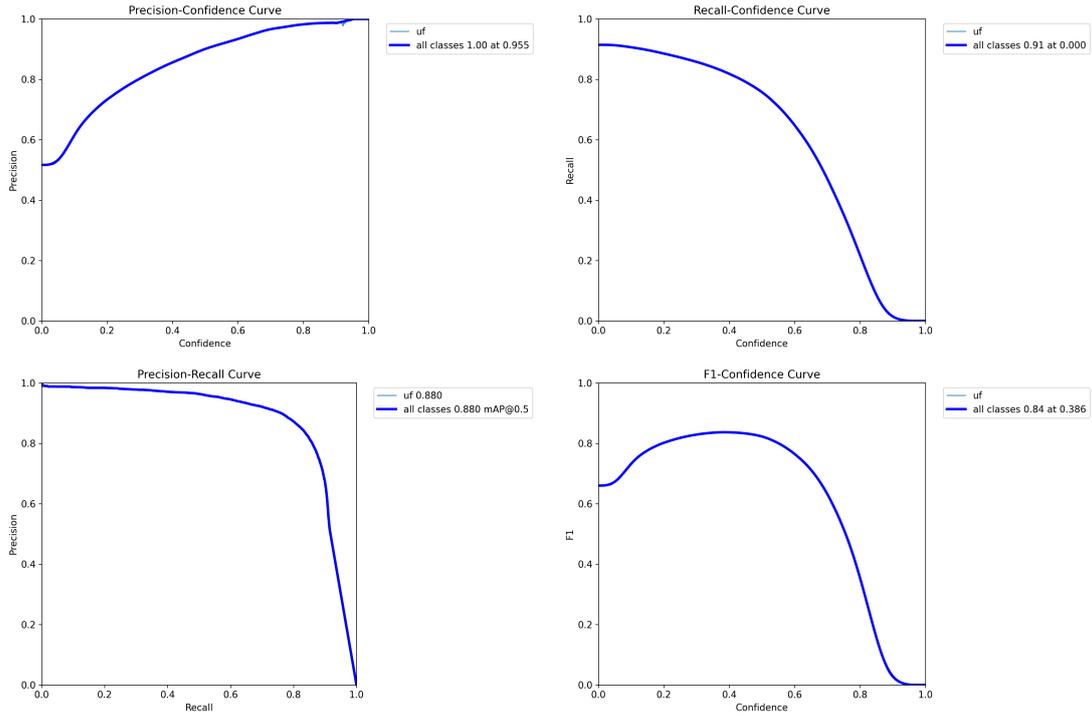
Após a aquisição de um conjunto adicional de imagens a cores, procedeu-se à realização de testes suplementares de detecção nessas mesmas imagens, isto usando os pesos do treino no qual as imagens do *dataset* eram todas a preto e branco. Os resultados obtidos revelaram um desempenho significativamente satisfatório. Conforme ilustrado na Figura 4.16, é evidente que praticamente todos os folículos foram identificados com precisão, apresentando níveis de confiança notavelmente elevados.



**Figure 4.16:** Detecções *YOLOv8* a Cores

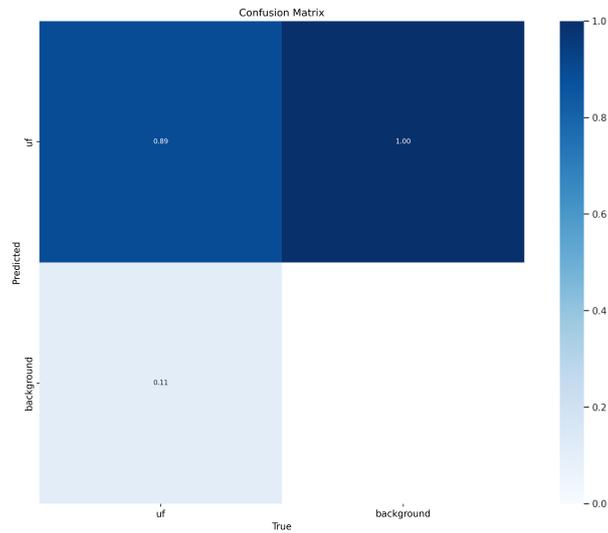
Ao observar os gráficos na Figura 4.17, é possível identificar tendências e padrões que revelam a sensibilidade das métricas à variação do limiar de confiança. Por exemplo, é notável como a precisão da detecção evolui à medida que o limiar de confiança é ajustado. Em alguns casos, um limiar de confiança mais alto pode resultar em detecções mais precisas, porém, em contrapartida, pode reduzir o número total de detecções. Por outro lado, um limiar de confiança mais baixo pode levar a um maior número de detecções, mas algumas delas podem ser menos confiáveis. Este exemplo está demonstrado na curva *F1-Confidence*.

É de referir que nos gráficos apresentados, verifica-se um comportamento ligeiramente melhorado do que os da rede *YOLOv5*, mostrados anteriormente na Figura 4.9.

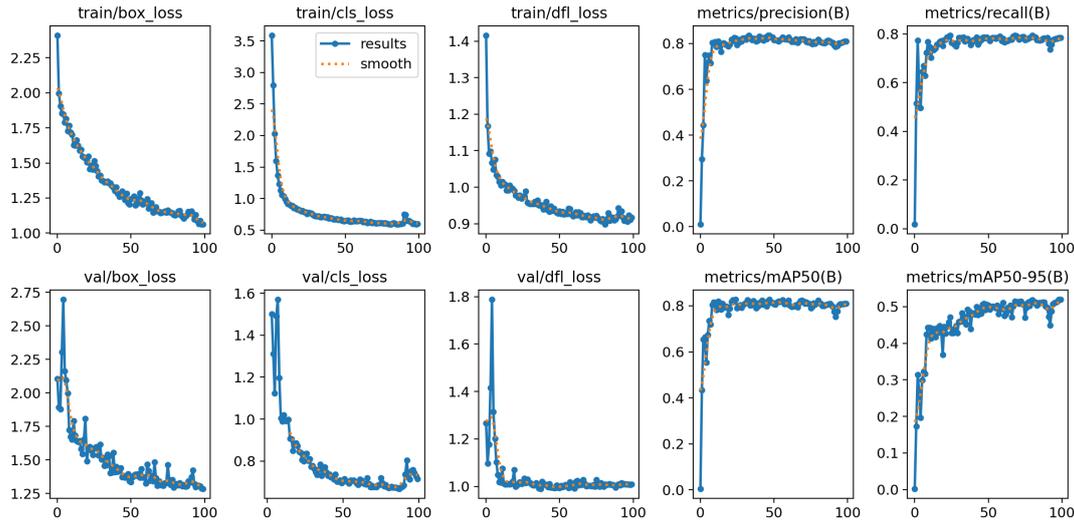


**Figure 4.17:** Métricas do Treino da Rede *YOLOv8*

A matriz de confusão está representada na Figura 4.18, pode-se verificar que, apesar de pequenas as diferenças com a rede *YOLOv5*, demonstram que o comportamento da rede ainda varia mesmo com um treino com os parâmetros de entrada iguais e com o mesmo conjunto de dados de treino.



**Figure 4.18:** Matriz de Confusão da Rede *YOLOv8*



**Figure 4.19:** Resultados da Rede *YOLOv8*

**Table 4.2:** Métricas Derivadas do Treino da Rede *YOLOv8*

	Valor(0-1)	Porcentagem(0-100)
<b>Precision</b>	0.81	81%
<b>Recall</b>	0.769	76.9%
<b>mAP50</b>	0.815	81.5%
<b>mAP50-95</b>	0.441	44.1%

Pelos resultados obtidos, é evidente que as métricas atingem valores significativamente altos, consoante o conjunto de dados utilizado para o treino.

Apesar de o *mAP 50-95* não apresentar um patamar tão elevado, as deteções de baixa confiança demonstram um nível de precisão notável. A partir dos valores obtidos através de ambas as redes *YOLOv5* e *YOLOv8* procedemos então aos testes de *tracking* utilizando estas redes.

#### 4.4 Teste *Tracking*

Dado que, todo o software da empresa é implementado em *ROS*, foi necessário criar um *script* à parte para iniciar os testes de *tracking*, de forma a não alterar significativamente a estrutura do código existente.

Assim, para dar início aos testes de *tracking*, foram utilizados os pesos das redes *YOLOv5* e *YOLOv8*, disponíveis na internet, com o recurso do conjunto de dados COCO. As primeiras tentativas foram conduzidas com vídeos que apresentavam automóveis em andamento no trânsito e pessoas a circular nas ruas. A seleção desses vídeos foi devido ao facto, de existir um treino prévio disponível, com resultados bastante promissores para essas classes específicas.

Após serem realizados os testes necessários para verificar se era viável a utilização do algoritmo SORT, foi iniciado o teste do mesmo, mas com os pesos obtidos através do treino efetuado pelas redes *YOLOv5* e *YOLOv8*.

A fim de viabilizar o processo de teste do *tracking* com as redes mencionadas anteriormente, foram desenvolvidos dois *scripts*.

O primeiro *script* foi criado para armazenar todas as imagens recebidas através de um tópico em ROS. As imagens são passadas através de um *bag*, e ao correr esse *bag*, ele fornece um conjunto de imagens para serem utilizadas na detecção das redes. Foi escolhido um conjunto de imagens o melhor possível sem grande variação de movimentos, para não haver uma grande quebra, na distância das *bounding boxes* detetadas entre *frames*.

O segundo *script* tinha como principal função utilizar as imagens obtidas anteriormente para criar um vídeo com uma taxa de 20 *frames* por segundo. Com esse vídeo, foi aplicada a detecção da rede YOLOv5/YOLOv8 com os pesos derivados do treino das redes mencionadas anteriormente.

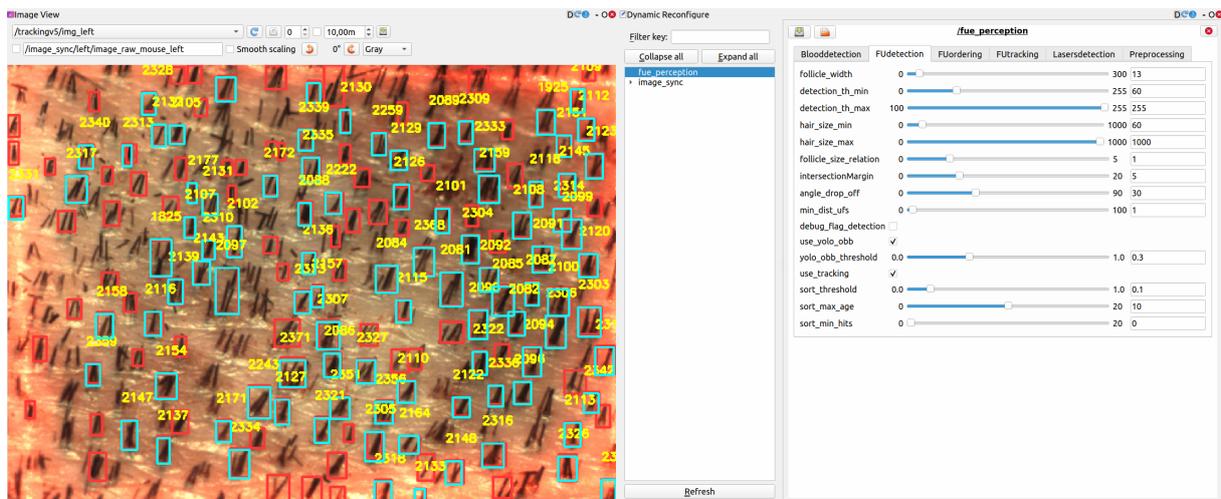
Após as detecções efetuadas em tempo real num vídeo, é necessário apenas aplicar o algoritmo SORT que depende das *bounding boxes* recebidas pelas redes neurais ou pelo software da empresa, tendo este que ter como parâmetros de entrada:

$$\langle x1, y1 \rangle, \langle x2, y2 \rangle, \langle conf \rangle$$

Cada um destes pontos tem o seguinte significado:

- $\langle x1, y1 \rangle$  Canto inferior esquerdo da *bounding box* detetada.
- $\langle x2, y2 \rangle$  Canto superior direito da *bounding box* detetada.
- $\langle conf \rangle$  Confiância da detecção em causa que varia entre 0 e 1.

Ao fazer a inferência da rede YOLO através do ficheiro "detect.py", são devolvidas como output, as coordenadas das respetivas *bounding boxes* dos folículos detetados, juntamente com o grau de confiança da detecção de cada *bounding box*. Esse processo viabiliza a aplicação direta do algoritmo SORT, uma vez que todos os dados essenciais para a sua execução são obtidos imediatamente a partir dos resultados gerados pela rede YOLO.



**Figure 4.20:** Tracking Output através das redes YOLOv5 e YOLOv8.

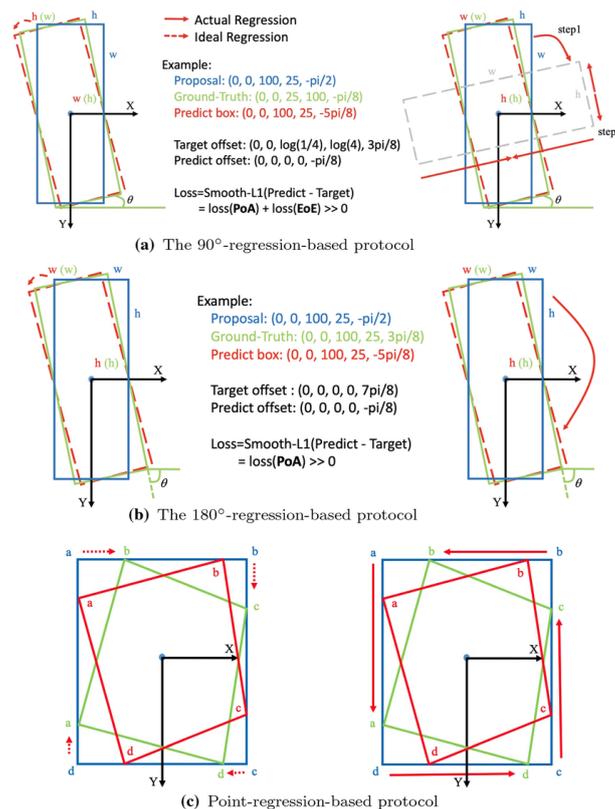
Através da Figura 4.20 é possível observar as detecções associadas a um ID, específico que

se mantém ao longo de um período de tempo até ser perdido. As detecções a vermelho são as novas detecções efetuadas pela rede YOLO que ainda não foram associadas a um ID pois não cumpriram os requisitos de *IoU* ("Interception Over Union"). Isto ocorre normalmente devido a mudanças bruscas entre *frames*, o que faz com que a unidade folicular se perca necessitando de um novo ID.

## 4.5 Rede detecção Orientada - *YOLOv5\_OBB*

### 4.5.1 Introdução à rede *YOLOv5\_OBB*

No âmbito da detecção de objetos, uma mudança paradigmática é introduzida pela modificação de rede YOLOv5, que permite a detecção orientada de objetos, a qual é designada de *YOLOv5\_OBB*, uma inovação que redefine a forma como os objetos são identificados em ambientes visuais complexos. Com um compromisso profundo em abordar as complexidades da orientação e escala dos objetos, a rede *YOLOv5\_OBB* introduz o conceito de *oriented bounding boxes (OBB)* para aprimorar a precisão e robustez das metodologias de detecção.



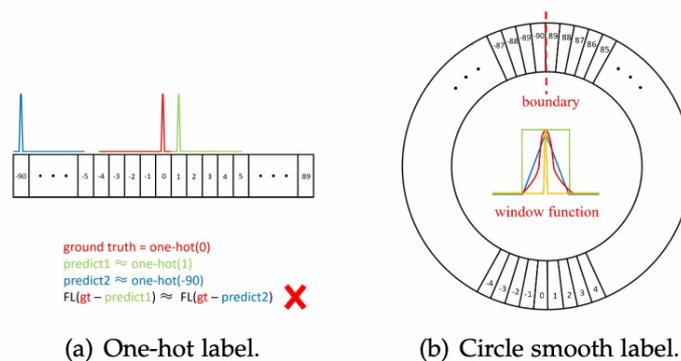
**Figure 4.21:** Ilustração para o problema de fronteira que persistentemente existe em 3 categorias populares de protocolos baseados em regressão. A seta vermelha sólida indica o processo de regressão real, e a tracejada vermelha mostra o processo de regressão ideal.[6]

No cenário da detecção com orientação, houve uma evolução notável na abordagem para obter previsões angulares precisas. Em contraste com os métodos convencionais de classificação, que muitas vezes enfrentam desafios na previsão de ângulos com alta precisão, surgem inovações

como a técnica *Circular Smooth Label (CSL)* [31]. A CSL destaca-se ao transformar o complexo problema de detecção de rotação num desafio de classificação, proporcionando vantagens cruciais.

A abordagem convencional baseada em regressão, exemplificada na Figura 4.21, tem suas limitações evidentes. A transição para a abordagem CSL revela-se como uma resposta perspicaz a esses desafios, abordando-os de forma abrangente. Destacando-se 2 fatores principais, tais como:

- Em primeiro lugar, é importante abordar o chamado "permutabilidade de arestas" (*EoE - exchangeability of edges*), que persiste mesmo quando se utiliza o protocolo baseado em  $90^\circ$  para *bounding boxes*. A peculiaridade do protocolo de  $90^\circ$ , que apresenta casos de contorno verticais e horizontais, contrasta com o protocolo de  $180^\circ$ , que se concentra exclusivamente em casos de contorno verticais. Isto cria uma complexidade adicional na previsão de ângulos.
- Segundo, a abordagem convencional de perda de classificação muitas vezes ignora a distância angular entre rótulos previstos e rótulos verdadeiros. Esse deslocamento da realidade angular prejudica a precisão necessária para o problema de previsão de ângulos. Num cenário em que o valor real seja  $0^\circ$ , e os resultados previstos pelo classificador sejam  $1^\circ$  e  $-90^\circ$ , respetivamente como demonstrado na Figura 4.22. Sob a abordagem convencional, ambas as previsões incorretas teriam a mesma perda, ignorando a importância de se aproximar do ângulo real de detecção.



**Figure 4.22:** Dois tipos de *labels* para classificação angular.[6]

Aqui entra o papel crucial da técnica CSL. Concebida para melhorar a previsão angular de forma sólida, esta adota a classificação como base, contornando as armadilhas das condições de contorno, incluindo o EoE e a "periodicidade angular" (PoA) [6]. É necessário ressaltar que CSL soluciona especificamente o problema da PoA, enquanto a questão do EoE encontra resolução no método de definição angular de  $180^\circ$ . Visualmente, a CSL abrange um sistema de codificação de rótulos circulares com periodicidade, o que confere suavidade aos valores atribuídos aos rótulos, considerando uma margem de tolerância.

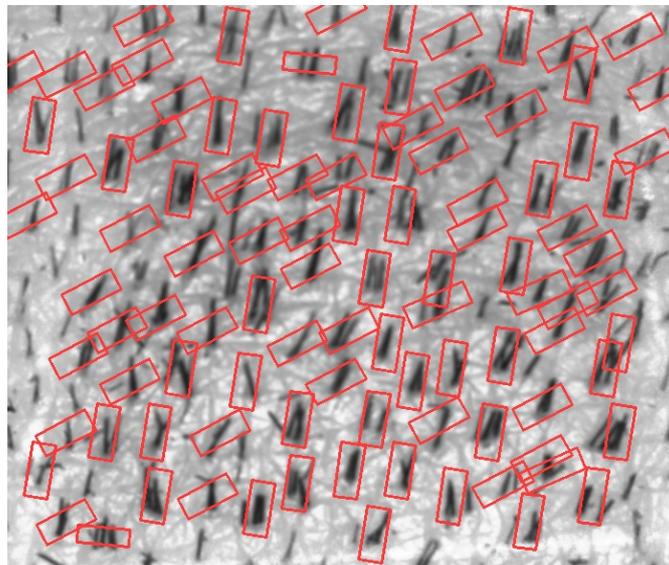
A rede *YOLOv5\_OBB* emerge como uma visão renovada, ao incluir os princípios inovadores da técnica *Circular Smooth Label (CSL)* na sua arquitetura. Isso, não somente permite a adaptação

do modelo *YOLOv5* para realizar uma deteção precisa, mas também habilita a rede a fornecer informações essenciais sobre a orientação dos objetos detetados. Embora esta rede tenha sido originalmente concebida para imagens aéreas capturadas por drones, a aplicabilidade dessa rede transcende esse contexto, e surge a pergunta: por que não aplicá-la na deteção de unidades foliculares?

Tendo como alvo a otimização da velocidade e confiabilidade na deteção das unidades foliculares, considerando a necessidade da obtenção da sua orientação para uma subsequente extração, a decisão da escolha desta rede é derivada da sua rapidez na deteção e o facto de fornecer prontamente a orientação das unidades foliculares detetadas.

#### 4.5.2 Treinos Iniciais da Rede *YOLOv5\_OBB*

Foi iniciado o processo de treino da rede utilizando um conjunto de dados inicial bastante reduzido, composto por imagens exclusivamente a preto e branco, cedidas inicialmente pela empresa para um teste preliminar. O resultado desse período de avaliação é exemplificado na Figura 4.23. Embora os resultados obtidos tenham tido sucesso na identificação correta da posição (x,y) do centro das unidades foliculares, a proporção das identificações corretas com a devida orientação foi bastante reduzida. No entanto, é importante salientar que as deteções realizadas estavam corretas em termos de identificação, embora a orientação precisa não tenha sido completamente alcançada.

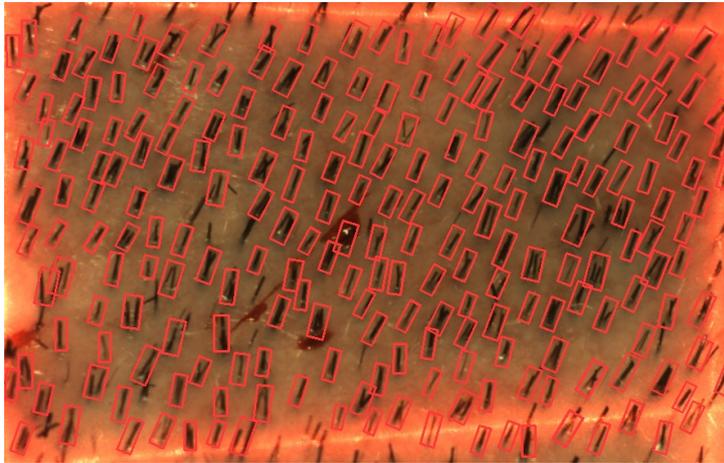


**Figure 4.23:** Treinos Iniciais da Rede.

A partir das deteções corretas obtidas mesmo com um conjunto de dados limitado, foi tomada a decisão de expandir o nosso *dataset* para um melhoramento gradual do treino. O processo foi começado por adicionar imagens, agora a cores, englobando diversas orientações e dimensões das unidades foliculares. Esse procedimento resultou em avanços promissores, embora ainda se percebesse uma certa quantidade de ruído presente nos resultados.

Tendo então este novo conjunto de dados melhorado, foi procedida a sua divisão de maneira

aleatória, reservando 70% para o treino do modelo e os restantes 30% para fins de validação e teste, ficando com 274 imagens para treino e 119 imagens para validação.



**Figure 4.24:** Treinos Finais da Rede.

Foram removidas algumas imagens que pareciam estar a interferir com a validação para obter umas métricas, o mais próximo possível da realidade, e obtemos então os seguintes resultados derivados dos treinos iniciais da rede.

**Table 4.3:** Parâmetros e Resultados Iniciais

Treinos	Optimizer	Batch Size	Epochs	Image Size	Precision-Recall (%)
Treino 1	adam	16	300	352	49.1 - 46.5
Treino 2	adam	16	400	352	53.3 - 55.0
Treino 3	adam	8	500	480	55.2 - 53.8
Treino 4	adam	32	500	192	11.0 - 09.0
Treino 5	sgd	16	500	352	05.0 - 06.7

Pelos resultados obtidos, foi possível verificar com que valores de *batch size* e *image size* a rede se comportava melhor, para depois no treino final ter em conta estes valores. Foi testado várias vezes o otimizador sgd mas, não foi possível a obtenção de valores favoráveis. Os resultados não foram excelentes em termos de métricas, até porque o conjunto de validação não estava excelente, tinha bastante ruído nas imagens de validação, então foi procedida a remoção de algumas imagens, e a sua passagem para o conjunto de treino devido às poucas imagens existentes. Os resultados com o novo conjunto de treino serão apresentados mais à frente, mas por observação destes valores, já é possível ter uma ideia como a rede se comporta em termos dos parâmetros de entrada.

Não foi possível realizar treinos com um *batch size* e *image size* superiores a 8-480 e 16-358 respetivamente, devido à falta de memória da GPU e isto poderia ser um ponto para trabalho futuro.

### 4.5.3 Tempos de Computação

Na Figura 4.25, está apresentado um exemplo dos tempos de processamento de detecção de unidades foliculares do software da empresa Insparya em comparação com os tempos de detecção observados pela rede *YOLOv5\_OBB*. O conteúdo da Figura 4.25 é derivado dos resultados de output da consola ao correr o ambiente ROS, na parte superior está apresentado o output da rede *YOLOv5\_OBB* e na parte inferior a inferência efetuada pelo software da empresa.

Através desta representação, é possível visualizar e comparar de forma clara e concisa os tempos de processamento entre as diferentes abordagens.

```
=====
YOLO INFERENCE:
Received an Image!

Time of Inference: 0.0058133602142333984
Time of NMS: 0.001275777816772461
Time of Processing Predictions: 0.01894354820251465

Found [137] UF
UF Detection Total Time: 0.02767491340637207
=====
INFO] [1693342793.249152725]: -----
INFO] [1693342793.249210127]: -----START-----
INFO] [1693342793.249219249]: -----
INFO] [1693342793.261709737]: Blood detection time: 0.012453
INFO] [1693342793.261736461]: -----
INFO] [1693342793.327536764]: Find laser lines: 0.065767
INFO] [1693342793.327564838]: -----
INFO] [1693342793.350862841]: Pre-processing time: 0.023277
INFO] [1693342793.350888692]: -----
INFO] [1693342793.372168205]: UF detection time: 0.021246
INFO] [1693342793.399183845]: Total time was: 0.149923
=====
```

**Figure 4.25:** Comparação Tempos de Detecção.

Com o objetivo de adquirir uma compreensão mais aprofundada dos tempos médios essenciais para conduzir uma detecção através da rede *YOLOv5\_OBB*, juntou-se um conjunto de dados relevantes relacionados com os tempos dos pontos cruciais da sua utilização. Estes dados englobam a fase de inferência, o tempo necessário para a aplicação do *NMS (Non-Maximum Suppression)* e o processo final de processamento.

Na Tabela 4.4, estão apresentados um conjunto abrangente de dados que oferece uma perspetiva detalhada sobre os tempos médios requeridos para cada fase do processo de detecção numa única imagem. Estes números fornecem uma visão concreta das diferentes etapas envolvidas e ajudam a esclarecer as discrepâncias temporais entre as abordagens avaliadas. Ao analisar os valores apresentados nesta tabela, é possível discernir claramente os períodos gastos em cada etapa específica, o que contribui para uma compreensão mais profunda da dinâmica de execução da rede.

Na Tabela 4.5, foi realizada uma análise comparativa abrangente dos tempos de detecção total. Esta medida é obtida pela soma dos tempos de inferência, o período necessário para aplicar o *NMS (Non-Maximum Suppression)* e a etapa final de processamento. Esses valores são confrontados com os tempos de detecção do software utilizado pela empresa Insparya.

Ao compreender os tempos envolvidos em cada componente, é necessário avaliar e otimizar o desempenho global do sistema, além de identificar potenciais áreas de melhoria. A Tabela 4.5

**Table 4.4:** Tempos de Inferência da rede *YOLOv5\_OBB* em segundos

Images	Inference (s)	NMS (s)	Process Predictions (s)
<b>1</b>	0.00575	0.00169	0.01873
<b>2</b>	0.00577	0.00278	0.02733
<b>3</b>	0.00577	0.00211	0.03637
<b>4</b>	0.00581	0.00137	0.01798
<b>5</b>	0.00574	0.00159	0.01774
<b>6</b>	0.00572	0.00265	0.02570
<b>7</b>	0.00573	0.00182	0.01791
<b>8</b>	0.00656	0.00337	0.02833
<b>9</b>	0.00577	0.00170	0.02304
<b>10</b>	0.00783	0.00334	0.02838

oferece uma visão clara e comparativa desses intervalos temporais, facilitando a análise criteriosa dos resultados obtidos na detecção das imagens do couro cabeludo.

**Table 4.5:** Tabela de Comparação dos Tempos de Detecção Total das Imagens em segundos

Images	Insparya Detection Time	<i>YOLOv5_OBB</i> Detection Time	Time Difference
<b>1</b>	0.011450	0.029952	0.018502
<b>2</b>	0.010867	0.028935	0.018068
<b>3</b>	0.010730	0.030544	0.019814
<b>4</b>	0.010442	0.027674	0.017232
<b>5</b>	0.009437	0.030342	0.020905
<b>6</b>	0.009958	0.031222	0.021264
<b>7</b>	0.010582	0.026579	0.015997
<b>8</b>	0.010511	0.029557	0.019046
<b>9</b>	0.009449	0.026900	0.017451
<b>10</b>	0.010285	0.028652	0.018367
<b>11</b>	0.010394	0.029710	0.019316
<b>12</b>	0.010542	0.027161	0.016619
<b>13</b>	0.010483	0.027786	0.017303
<b>14</b>	0.010409	0.023516	0.013107
<b>15</b>	0.009860	0.029204	0.019344
<b>16</b>	0.010051	0.025031	0.014980
<b>17</b>	0.010742	0.023256	0.012514
<b>18</b>	0.009827	0.023091	0.013264
<b>19</b>	0.010618	0.023961	0.013343
<b>20</b>	0.010742	0.030991	0.020249

É essencial ressaltar que os valores presentes na Tabela 4.5 não se correlacionam com as imagens mencionadas na Tabela 4.4. Estes valores são obtidos através da inferência da rede *YOLOv5\_OBB* e do software de detecção da empresa Insparya na mesma imagem, possibilitando uma comparação pertinente.

Na Tabela 4.6 estão apresentadas as médias consoante os valores da Tabela 4.5, isto para

termos um meio termo de comparação.

**Table 4.6:** Tabela das Médias de Comparação dos Tempos de Detecção das Imagens em segundos

	<b>Insparya Detection Time</b>	<b>YOLOv5_OBB Detection Time</b>	<b>Time Difference</b>
<b>Média</b>	0.010499	0.028085	0.017587

Uma vez obtidos os valores mencionados anteriormente, é possível agora proceder com as análises necessárias para quantificar a eficiência da rede *YOLOv5\_OBB* em comparação com o software desenvolvido pela Insparya.

Para alcançar essa compreensão, é crucial realizar cálculos que permitam determinar a diferença de velocidade entre os dois métodos. Ao dividir o tempo de detecção da *YOLOv5\_OBB* pelo tempo de detecção da Insparya, obtemos um coeficiente que nos indica o quão mais rápida é a última em relação à primeira.

$$\text{Velocidade} = \frac{0.028085}{0.010499} = 2.68 \quad (4.1)$$

É possível observar que, a partir dos dados apresentados na Tabela 4.5, emerge alguma discrepância entre os valores de tempo de detecção obtidos pelo software da empresa Insparya e os tempos de detecção apresentados pela rede *YOLOv5\_OBB*. Esta discrepância é evidente nas várias instâncias analisadas, onde os valores de tempo de detecção da empresa Insparya são ligeiramente inferiores aos tempos correspondentes da *YOLOv5\_OBB*.

Essa discrepância reflete-se na coluna de "*Time Difference*", que evidencia as diferenças entre os tempos de detecção das duas abordagens. Os valores presentes nessa coluna realçam a diferença de desempenho. Pelos valores apresentados podemos concluir que, o tempo médio de detecção do software da empresa corresponde a 2.68 vezes ao tempo de detecção da rede *YOLOv5\_OBB* como mostrado na equação 4.1, sabendo que, a isto acresce ainda o tempo de passagem de parâmetros. Os tempos de computação da rede, mesmo sendo superiores aos valores obtidos pelo software de detecção da empresa, estão dentro dos valores aceitáveis para o resto da computação necessária.

Ao comparar os tempos de detecção para diferentes conjuntos de imagens, é possível observar consistentemente que a rede *YOLOv5\_OBB* oferece tempos de detecção ligeiramente superiores em comparação com os do software da Insparya. Essa disparidade demonstra uma pequena desvantagem em termos de eficiência de detecção da rede *YOLOv5\_OBB*, mas pode ser uma vantagem se compensar pela precisão e robustez na detecção.

Esta conclusão é importante para futuras otimizações e ajustes, orientando a abordagem a ser tomada para aprimorar o desempenho do sistema e reduzir os tempos de detecção das unidades foliculares.

#### 4.5.4 Melhoramento do *dataset* (Novos Treinos)

Depois de adquiridas mais imagens, através das anotações feitas pelos enfermeiros da empresa, foram realizados treinos adicionais da rede *YOLOv5\_OBB*, tendo em consideração os resultados

obtidos anteriormente. Foram incluídas apenas algumas imagens no conjunto de validação, as melhores consideradas para ser feita a detecção em si, sem muito ruído e interferências, para existir um bom termo de comparação com o software de detecção da empresa Insparya.

Devido ao facto de o número de imagens para o treino serem reduzido e também à falta de tempo dos enfermeiros para anotar imagens e à dificuldade de obter imagens novas para incluir no *dataset*, pois apenas podem ser obtidas através da operação de pacientes. Isto, foi o que induziu a utilizar o máximo número de imagens no treino para existir a maior variedade possível e então a divisão do *dataset* foi feita da seguinte maneira: 480 imagens para treino e 47 imagens para a validação. As imagens escolhidas para a validação foram tidas em conta relativamente ao software da empresa e a sua pouca facilidade de adaptação a ambientes diferentes e como temos uma maior variedade seria mais complicado para a sua detecção, então foram escolhidas as melhores imagens para ambos. Inicialmente o *dataset* possuía 340 imagens de treino e após mais algumas imagens obtidas ficamos com as 480.

Os treinos foram efetuados com *early stopping* de 100 épocas, o que significa que se a rede não apresentar melhorias durante 100 épocas, o treino termina guardando os melhores valores dos pesos obtidos durante o treino completo. Portanto, os valores apresentados das épocas na tabela contam com a paragem do *early stopping*.

**Table 4.7:** Treinos com *dataset* completo

Treinos	Optimizer	Epochs	Batch Size	Image Size	nº Imagens	Precision-Recall (%)
Treino 1	adam	331	8	480	340	62.2 - 64.0
Treino 2	adam	391	16	352	340	56.7 - 59.7
Treino 3	adam	500	8	480	340	51.6 - 55.9
Treino 4	adam	500	8	480	480	68.7 - 74.5
Treino 5	adam	767	8	480	480*	70.2 - 74.0
Treino 6	adam	700	8	480	480*	70.7 - 74.4
Treino 7	adam	723	8	480	480*	70.7 - 75.4
Treino 8	adam	1037	16	352	480*	65.1 - 59.9

Na Tabela 4.7 estão apresentados os valores respetivos dos melhores treinos efetuados com a rede *YOLOv5\_OBB* após serem adicionadas novas imagens e o *dataset* ter sido reformulado, isto para ter noção dos parâmetros que mais influenciaram os resultados dos treinos da rede.

No Treino 3 apresentado na Tabela 4.7 os valores são inferiores devido ao aumento do *threshold* de IoU para 0.6 utilizado para a validação do treino efetuado.

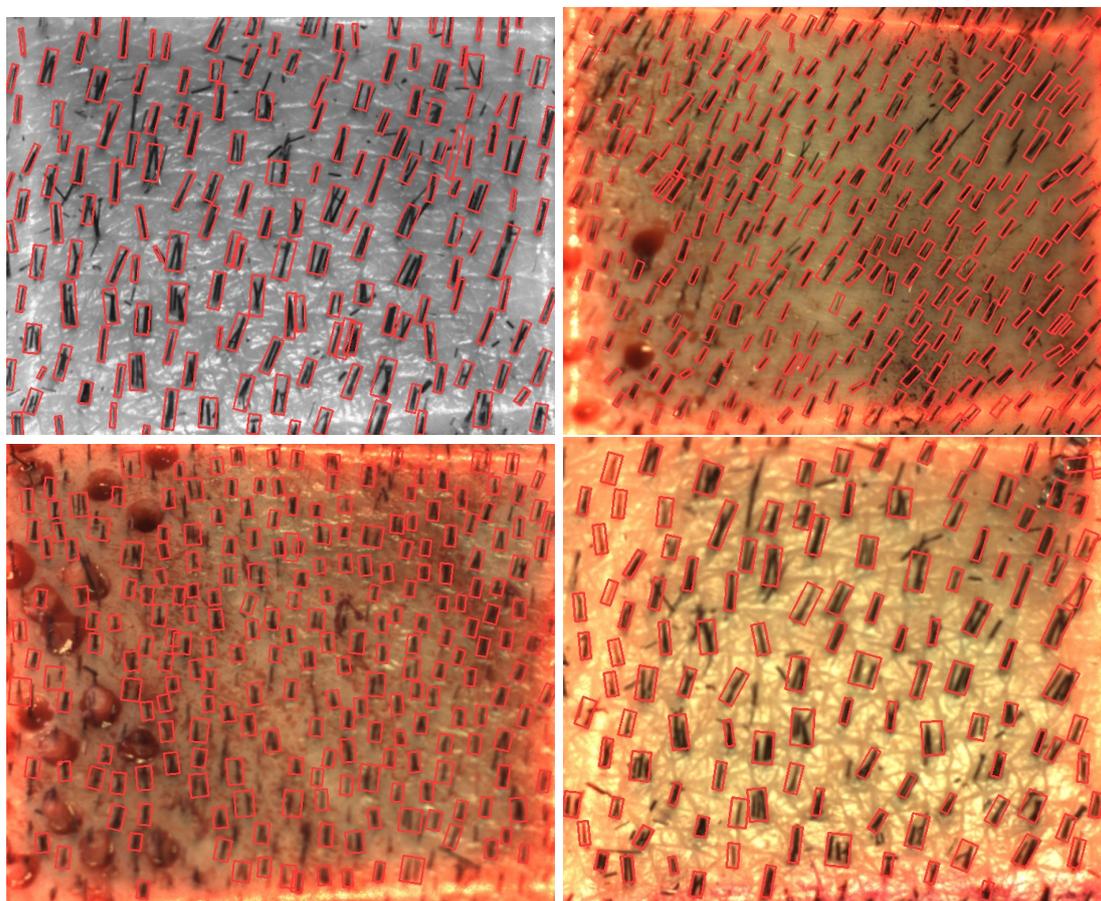
Relativamente aos treinos que contêm um "\*" no número de imagens, da Tabela 4.7, significa que foi utilizada a *data augmentation* presente na rede, tem um parâmetro que é a probabilidade de "mosaic" e ao colocar este parâmetro a 1, ou seja 100%, ele cria por cada imagem do *dataset* de treino mais 3 imagens alteradas, a partir da original, segundo os parâmetros de *data augmentation* utilizados, ficando assim com um *dataset* 4 vezes maior, 1920 imagens. Como a variedade de imagens não é muita, foram alterados bastantes parametros para a *data augmentation*, como por exemplo o brilho das imagens, *flip* vertical e horizontal, rotações, translações, perspectiva,

saturação, entre outros. Todos estes procedimentos foram feitos para existir a maior variedade possível de imagens e aumentar o número de imagens de *dataset* de treino.

Os hiper-parâmetros utilizados no melhor treino da rede foram os seguintes:

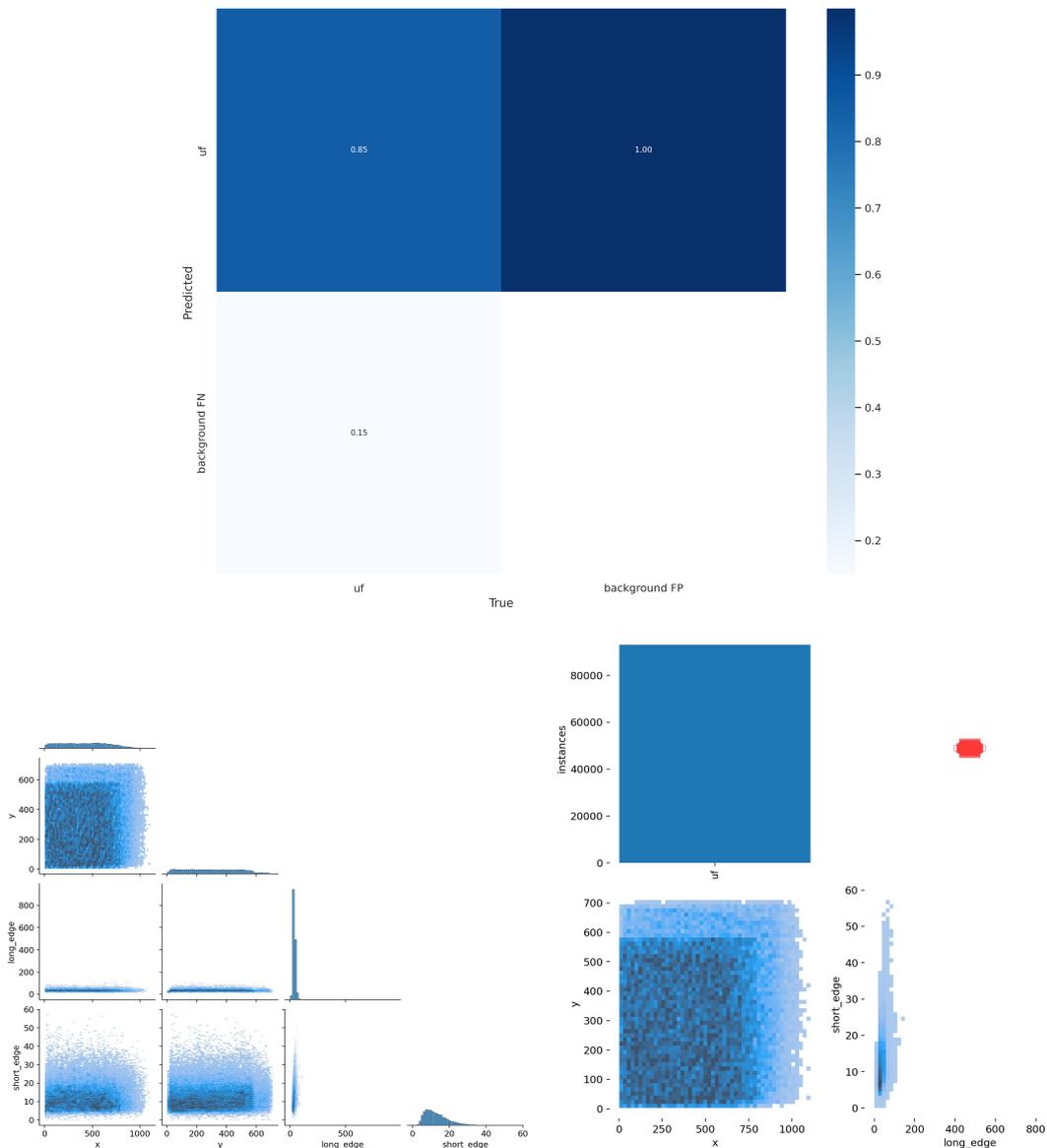
```
1 lr0: 0.00258
2 lrf: 0.17
3 momentum: 0.779
4 weight_decay: 0.00058
5 warmup_epochs: 1.33
6 warmup_momentum: 0.86
7 warmup_bias_lr: 0.0711
8 box: 0.0539
9 cls: 0.299
10 cls_pw: 0.825
11 theta: 0.299
12 theta_pw: 0.825
13 obj: 0.632
14 obj_pw: 1.0
15 iou_t: 0.2
16 anchor_t: 3.44
17 anchors: 3.2
18 fl_gamma: 0.0
19 hsv_h: 0.0188
20 hsv_s: 0.704
21 hsv_v: 0.36
22 degrees: 180.0
23 translate: 0.0902
24 scale: 0.25
25 shear: 0.0
26 perspective: 0.0
27 flipud: 0.5
28 fliplr: 0.5
29 mosaic: 1.0
30 mixup: 0.0
31 copy_paste: 0.0
32 cls_theta: 180
33 csl_radius: 2.0
```

**Figure 4.26:** Hiper-Parâmetros utilizados no Treino 7 da Tabela 4.7.



**Figure 4.27:** Deteções efetuadas com o Treino 7 da Tabela 4.7.

Através da visualização das detecções efetuadas pela rede *YOLOv5\_OBB*, representadas na Figura 4.27 é possível ver quase todos os folículos bem detetados, com a falta de alguns na zona dos orifícios que representam unidades foliculares já extraídas. Como naquela zona existe uma grande quantidade de ruído é mais complicado, mas é possível fazer um treino mais aprofundado nessas zonas, mas para isso seria necessário mais imagens, o que serve para trabalho futuro. É importante salientar que, ao detetar unidades foliculares na imagem, o foco principal está nos folículos centrais. Essa, é a região onde a ferramenta de extração será aplicada. Neste ponto, a métrica de *Precision* é extremamente importante para assegurar que as unidades foliculares detetadas estejam localizadas com precisão, minimizando o risco de qualquer lesão acidental ao paciente com a ferramenta de extração, evitando a aplicação da ferramenta em áreas onde não há unidades foliculares presentes, ou seja em Falsos Positivos.



**Figure 4.28:** Matriz de Confusão e Correlograma das Labels Provenientes do Treino 7 da Rede *YOLOv5\_OBB*.

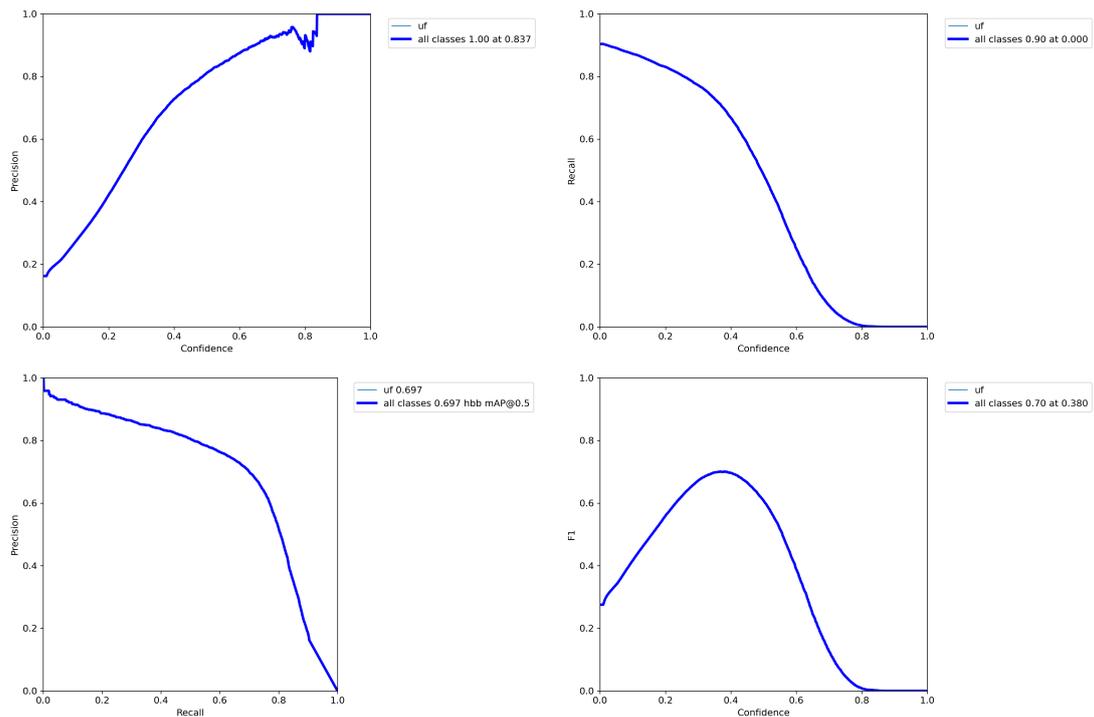


Figure 4.29: Curvas das Métricas Provenientes do Treino 7 da Rede *YOLOv5\_OBB*.

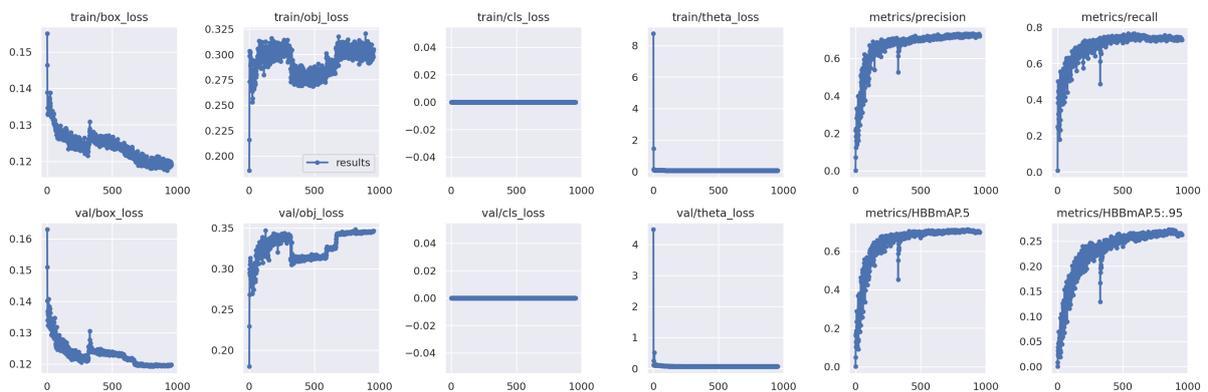


Figure 4.30: Resultados Provenientes do Treino 7 da Rede *YOLOv5\_OBB*.

Nas Figuras 4.28, 4.29 e 4.30, estão apresentados os gráficos e diagramas obtidos através do treino da rede *YOLOv5\_OBB* segundo o Treino 7 da Tabela 4.7.

#### 4.5.4.1 Métricas Empresa

O algoritmo da Empresa Insparya utiliza diversas funções do *OpenCV* que auxiliam na detecção das unidades foliculares, tais como:

- *Blackhat*: A operação de *Blackhat* é uma técnica de processamento de imagem que realça as características escuras (ou seja, as unidades foliculares) em relação ao fundo da imagem. É útil para melhorar a detecção de objetos de interesse, como os folículos capilares.
- *Threshold*: A função de *thresholding* é usada para segmentar a imagem, convertendo-a

numa imagem binária, onde os pixels de interesse (folículos) são destacados em branco e o fundo é representado em preto. Escolher o valor de *threshold* adequado é crucial para separar os folículos do restante da imagem.

- *findContours*: Após a binarização da imagem, a função *findContours* é usada para identificar os contornos dos objetos brancos na imagem. Esses contornos são essenciais para a análise subsequente, permitindo determinar a forma e a localização dos folículos.
- *Filter based on area and shape*: Depois de identificar os contornos dos folículos, são aplicados filtros para eliminar detecções indesejadas ou falsos positivos. Isto é feito com base em critérios como área (tamanho) e forma dos contornos. É possível definir um tamanho mínimo e máximo para os folículos com base na sua área e, se necessário, aplicar critérios de forma para garantir que apenas os folículos reais sejam detetados.

Através destas funções é possível obter detecções orientadas das unidades foliculares, mas para existir um método quantitativo de avaliação e não apenas visual, foi criado um código que avalia as métricas essenciais do software de detecção da empresa Insparya.

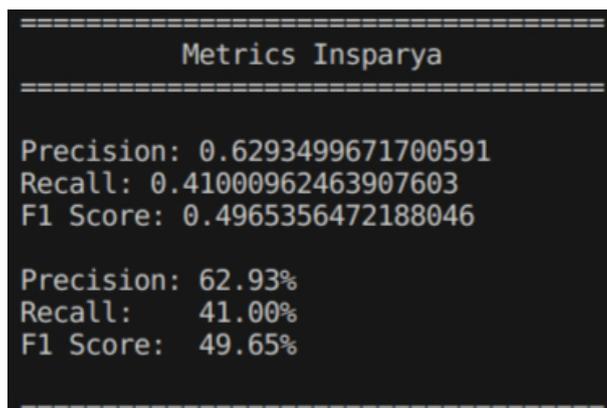
A arquitetura *YOLOv5\_OBB* fornece de forma direta as métricas essenciais para avaliação. No entanto, o software desenvolvido pela empresa não disponibiliza essas métricas diretamente. Isso levou à criação de um *script*, que desempenha um papel crucial ao permitir a comparação dos resultados obtidos entre a rede e o software da empresa. Esse *script* foi desenvolvido para avaliar os resultados produzidos pelo software de detecção da empresa usando um conjunto de dados especialmente selecionado para validação.

Foi publicado um tópico com exatamente as mesmas imagens que foram utilizadas para a validação do treino da rede, isto para serem detetadas pelo software de detecção da empresa e devolverem a cada uma das imagens, as respectivas detecções das unidades foliculares. Assim, cada imagem detetada tem as *labels* respectivas da detecção efetuada e as *labels ground truth* correspondentes, sendo assim possível fazer a comparação através da IoU e obter as métricas necessárias para existir um termo de comparação com os resultados da rede.

Primeiramente, foi criado um *script* que calculava a IoU, mas de *bounding boxes* orientadas, ou seja, fazia a área de um retângulo que continha os valores máximos e mínimos de (x,y), tendo assim a altura e largura de um retângulo sem orientação. Mas esses valores eram melhores do que o esperado, pois as *bounding boxes* eram maiores e havia uma maior sobreposição das mesmas, tendo assim um IoU mais elevado, contando com mais *True Positives* que não deveriam ser contados.

Após ser observada essa ocorrência, passou-se à criação de um segundo *script* que calculava a área do retângulo orientado, tendo assim umas métricas mais verdadeiras e próximas da realidade.

Foi utilizada a biblioteca "*sklearn metrics*" para o cálculo dos *True Positives* e *False Positives*, basicamente sempre que a área de sobreposição com o *ground truth* era superior a um certo threshold, era adicionado 1 ao conjunto dos *True Positives*, sempre que era menor ou mesmo 0 era adicionado 1 ao conjunto dos *False Positives*. [32]

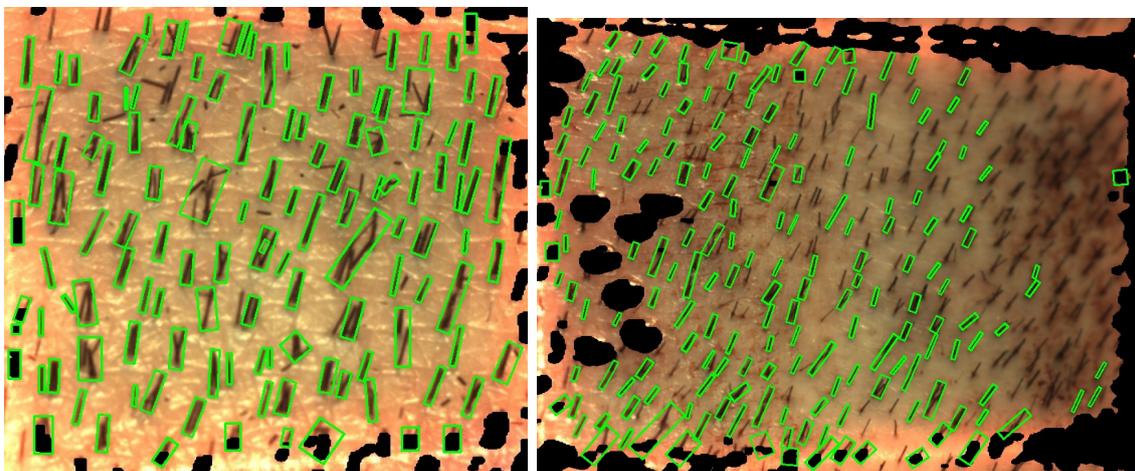


**Figure 4.31:** Métricas Obtidas Através das Deteções Efetuadas Pelo Software da Empresa

Na Figura 4.31 é possível observar as métricas obtidas através das deteções efetuadas pelo software de deteção da empresa com os mesmos dados de validação da rede *YOLOv5\_OBB*.

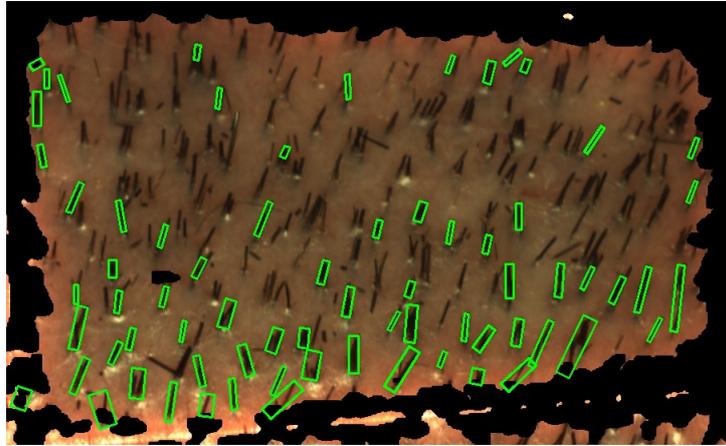
As métricas foram calculadas com um *threshold* de 0.4 para a IoU, que é um parâmetro que está predefinido pela rede, também usado nos treinos anteriores. Neste aspeto, o comportamento da rede é melhor, comparativamente ao software da empresa Insparya, claro que se os parâmetros forem ajustados pré-operação de acordo com o ambiente em questão, para um conjunto de imagens semelhantes, o software da empresa tem um comportamento muito bom, mas na ocorrência de uma interferência a meio, como vazamento de sangue ou desfoque da câmara ou menos luminosidade, a rede será uma melhor opção para estas situações. Os valores de *thresholds* definidos pelo software da empresa variam entre ambientes, saí ser necessário o ajuste pré-operação até serem obtidos os resultados de deteção esperados. Existem vários parâmetros tais como: largura da unidade folicular, *threshold* máximo e mínimo da deteção, tamanho máximo e mínimo das unidades foliculares, margem de interceção, relação do tamanho das unidades foliculares, mínimo de distância entre unidades foliculares e a queda angular.

O software da empresa tem imagens do conjunto de validação, que deteta totalmente, mas outras, com os mesmos parâmetros, deteta poucos folículos, daí, estes baixos valores das métricas apresentadas na Figura 4.31.



**Figure 4.32:** Exemplo Deteção com Software de Deteção da Empresa Insparya

As imagens apresentadas na Figura 4.32 representam as detecções efetuadas com o software da empresa Insparya. Na segunda imagem, é possível verificar que, em zonas sujas com sangue derivado da extração, ou zonas menos luminosas, não apresenta muita precisão nos resultados.



**Figure 4.33:** Exemplo Detecção com Software de Detecção da Empresa Insparya

Na Figura 4.33 está representada uma imagem também pertence ao conjunto de validação, anotada apenas pela metade, devido à proximidade das unidades foliculares. Será também feita uma comparação relativa às validação com os diferentes valores de *threshold* de IoU com as validações com os *thresholds* correspondentes da rede *YOLOv5\_OBB*.

## 4.6 Recriação 3D

A recriação 3D é um processo sofisticado que aproveita a precisão das câmaras calibradas em conjunto com a informação obtida, a partir de pontos estrategicamente selecionados nas imagens do objeto de interesse. Neste contexto, consideremos um exemplo prático em que aplicamos a recriação 3D para realizar uma tarefa específica, como manipular uma ferramenta num ambiente controlado.

Primeiramente, as câmaras calibradas desempenham um papel crucial. São configuradas de forma a capturar imagens com alta precisão e consistência. Em seguida, através das detecções efetuadas com a rede *YOLOv5\_OBB*, são escolhidos dois "*key points*" em cada imagem: um ponto na base e outro no topo de um folículo. A partir desses pontos, o sistema é capaz de realizar a triangulação, um processo matemático, que permite calcular a posição exata no espaço tridimensional desses pontos em relação às câmaras.

Com essas informações de triangulação, torna-se possível calcular vetores no mundo tridimensional que descrevem as direções e posições relativas dos pontos identificados. Essa etapa é fundamental para entender a geometria e a disposição do objeto no espaço.

Uma vez que temos essa compreensão do ambiente tridimensional, a próxima etapa envolve posicionar o robô. O objetivo é alinhar a ferramenta do robô de maneira a que a sua orientação seja a mais próxima possível da configuração anterior. Isto é crucial, para garantir que a ferramenta esteja na posição adequada para realizar a extração de uma unidade folicular.

Além disso, a ponta da ferramenta é posicionada no topo do folículo com base nas informações obtidas através do processo de recriação 3D. Essa abordagem garante que a ferramenta esteja alinhada com o objeto de interesse de maneira altamente precisa.

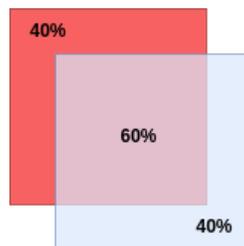
Finalmente, um eixo externo entra em ação para executar o movimento necessário. Nesse caso, o eixo é responsável por fazer a ferramenta avançar exatamente 5mm, que é a medida necessária para a remoção da unidade folicular.

## 4.7 *Tracking* com a deteção orientada

A interligação das novas tecnologias dos dias de hoje torna possível criar ferramentas com uma utilidade extrema. Neste caso, para fazer o seguimento das unidades foliculares, basta aplicar o algoritmo SORT às deteções derivadas da rede *YOLOv5\_OBB*. Esta estratégia atribui um identificador(ID) a cada folículo ao longo dos *frames*, permitindo um acompanhamento preciso e contínuo das trajetórias foliculares ao longo do tempo.

O algoritmo SORT, é conhecido pela sua eficiência, sendo capaz de lidar com a dinâmica em constante mudança das deteções e fazer o *tracking* de objetos em movimento, de forma consistente. Ao atribuir um ID exclusivo a cada deteção inicial, o sistema estabelece uma referência identificável para um folículo específico.

Através da utilização da métrica de sobreposição de área (IOU), que avalia a interseção entre as deteções atuais e a percentagem de área compartilhada com as deteções correspondentes do *frame* anterior, é possível estabelecer a proximidade entre deteções subsequentes e aquelas já previamente identificadas. Esta técnica viabiliza um acompanhamento coerente das trajetórias individuais de cada unidade folicular, mesmo quando enfrentando desafios como a oclusão parcial ou temporária de um folículo. Essa abordagem assegura uma continuidade no processo de *tracking*, abrindo caminho para uma análise minuciosa das alterações e padrões característicos de cada unidade folicular.



**Figure 4.34:** Exemplo IoU.

A Figura 4.34 ilustra um caso exemplar de Índice de Sobreposição de Área (IoU), que quantifica a proporção da sobreposição das áreas entre objetos detetados. Ao ajustar um *threshold* definindo a percentagem de sobreposição, é possível melhorar a precisão da atribuição de identificadores (IDs) aos *frames* correspondentes.

Dado que os parâmetros de entrada no algoritmo SORT seguem o seguinte formato de entrada:

$$\langle x1, y1 \rangle, \langle x2, y2 \rangle, \langle score \rangle$$

- $\langle x1, y1 \rangle$ : Coordenadas do canto superior esquerdo da *bounding box*.
- $\langle x2, y2 \rangle$ : Coordenadas do canto inferior direito da *bounding box*.
- *score*: Grau de confiança da resultante da detecção.

Sendo que os parâmetros de saída da rede *YOLOv5\_OBB* são as coordenadas (x, y) de todos os vértices de cada uma das *bounding boxes* orientadas devolvidas após a inferência numa imagem, juntamente com o grau de confiança da detecção correspondente, foi necessário realizar uma adaptação na organização dos parâmetros de saída da rede. Foi utilizado o mesmo método que está representado na Figura 4.2, mas isto incluído diretamente no código de detecção da rede *YOLOv5\_OBB*. Fazemos um retângulo sem orientação com os valores máximos e mínimos de x e y, tendo assim os dois cantos necessários para a configuração de input do SORT. Tornando assim mais fácil fazer a IoU de cada detecção, pois uma unidade folicular com uma *bounding box* orientada é muito estreita, o que torna muito difícil a sobreposição com o *frame* seguinte e assim gera um mau *tracking* associado a cada uma dessas unidades foliculares, gerando cada vez mais IDs.

O algoritmo SORT para além do *threshold* de IoU tem mais 2 parâmetros, os quais podemos alterar de modo a obter um resultado mais preciso e eficaz. Estes são:

- *max\_age* - se um objeto não for detetado num determinado número de *frames* consecutivos, este será considerado perdido e removido do conjunto de objetos rastreados. O valor de *max\_age* é geralmente definido em termos do número de *frames*. Um valor maior para *max\_age* permite que o algoritmo seja mais tolerante com a perda temporária de um objeto, o que pode ser útil em situações em que os objetos estão sujeitos a oclusões frequentes ou movimentos erráticos.
- *min\_hits* - representa o número mínimo de detecções consecutivas que um objeto deve ter para ser considerado como um objeto rastreado válido. Isso significa que um objeto apenas será considerado rastreado, depois de ser detetado por um número mínimo de vezes seguidas. Esse parâmetro ajuda a evitar falsos positivos, ou seja, objetos que aparecem esporadicamente e não têm uma trajetória clara de detecções.

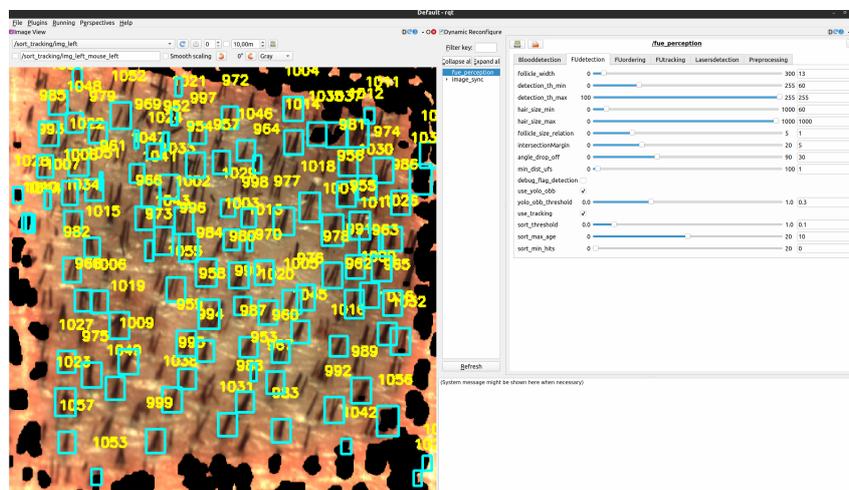


Figure 4.35: Tracking Utilizando a Rede *YOLOv5\_OBB*.

## 4.8 Interfaces

A incorporação completa da rede neuronal no ambiente *ROS (Robot Operating System)* estabelecido pela empresa desencadeou a oportunidade de desenvolver múltiplas interfaces. Estas são especialmente direcionadas para avaliar a atuação da rede *YOLOv5\_OBB* e do processo de *tracking*. As interfaces foram criadas utilizando a ferramenta *RQT (ROS Qt-based User Interface Toolkit)*.

Foram utilizados bags como meio para realizar as detecções. Ao inserir as imagens na rede por meio da execução de um *bag*, ocorre numa forma de *frame a frame*, permitindo que a rede faça as suas inferências de forma sequencial. Os resultados obtidos pela rede são então encaminhados para a publicação num tópico, o qual, por sua vez, transmite as informações para o RQT em tempo real, proporcionando uma análise mais aprofundada sobre o desempenho da rede permitindo a visualização de cada unidade folicular envolvida por uma *bounding box* correspondente.

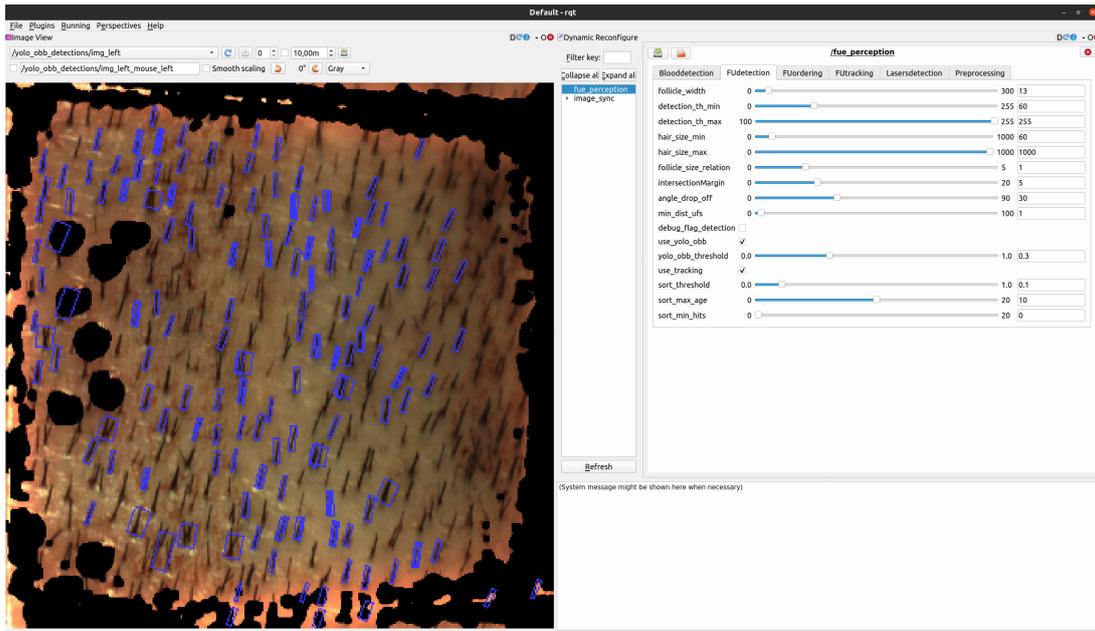
Outro aprimoramento significativo foi a introdução de um botão no *Dynamic Reconfigure*. Esse botão viabilizava a transição da utilização do software de detecção da empresa para a rede *YOLOv5\_OBB*. Ao desativar o botão, o sistema revertia para o software de detecção da empresa. Esse recurso conferiu uma flexibilidade notável às operações, possibilitando a rápida troca entre os métodos de detecção conforme a necessidade e o contexto, sem demandar uma intervenção complexa no sistema, permitindo assim uma avaliação visual entre os diferentes métodos.

Além disso, foi incluído também um controlo ajustável do *threshold* de confiança para as detecções realizadas através rede neuronal. Esta adição, visa alcançar o melhor resultado praticável em tempo real. Tal medida torna-se crucial para eliminar detecções incorretas com níveis de confiança muito baixos. A possibilidade de ajustar o *threshold* de confiança permite filtrar tais detecções indesejadas, melhorando assim a qualidade do processo de detecção e possibilitando uma avaliação precisa da situação em tempo real.

No contexto do *tracking*, também foi implementado um botão que permite alternar entre os estados de ativação e desativação do mesmo. Além disso, foram incorporados três controlos adicionais destinados a ajustar os parâmetros relevantes.

Para otimizar a eficácia do algoritmo SORT, foi inserido um controlo dedicado a cada um dos elementos de entrada essenciais. Esses controlos estão relacionados com parâmetros críticos do algoritmo, incluindo o *threshold* de *IoU (Intersect over Union)*, o valor máximo de idade (*max\_age*) e o número mínimo de detecções bem-sucedidas (*min\_hits*) necessárias para confirmar o *tracking*.

Estas adições não apenas permitem um controlo mais refinado do processo de *tracking*, mas também facilitam a personalização do algoritmo para se adequar às características específicas do ambiente e das tarefas em questão. Através desses controlos, é possível ajustar o desempenho do algoritmo de acordo com as necessidades variáveis, assegurando uma adaptação contínua e eficaz do *tracking* de objetos, para além do seu ajuste em tempo real, ajudando a obter o melhor resultado de *tracking* possível consoante o ambiente em questão.



**Figure 4.36:** Interface RQT.

Na Figura 4.36 temos representada a interface criada que ajudou a avaliar o comportamento de todo o software criado.



# 5

## Conclusão

Neste trabalho, foi abordado de forma abrangente o tema da detecção de unidades foliculares com sua devida orientação, utilizando redes neuronais como ferramenta central para esta tarefa. Neste sentido, foi possível explorar o potencial das redes neuronais na resolução de um desafio complexo no campo da medicina e estética capilar.

Comparar as abordagens utilizadas, que incluem o uso de redes neuronais e o software de detecção criado pela Insparya, desempenhou um papel fundamental na avaliação do potencial e na determinação do rumo da investigação futura, baseada no progresso alcançado até agora.

Adicionalmente, foi proposta a integração do *tracking* das unidades foliculares no sistema, permitindo um acompanhamento contínuo da trajetória de cada unidade folicular. Esta funcionalidade essencial proporcionou a cada unidade folicular um identificador único (ID), o que é crucial para a monitorização e análise em estudos clínicos e procedimentos médicos.

Nos próximos tópicos, serão consolidadas as conclusões obtidas ao longo deste trabalho, resumindo os principais resultados, desafios e contribuições, bem como destacando possíveis direções futuras de pesquisa neste domínio.

### 5.1 Comparação Redes

A comparação entre dois métodos de detecção é um elemento crucial no desenvolvimento desta tese, pois permite avaliar a eficácia e a adequação destes métodos em ambientes específicos. Quando se trata de comparar estes dois métodos, é essencial levar em consideração uma série de fatores, incluindo o treino da rede neuronal e os parâmetros do software de detecção da empresa ajustados para se adequarem ao ambiente em questão. Então, para isso, foi usado o conjunto de validação da rede *YOLOv5\_OBB* para ser detetado pelo método de detecção da empresa Insparya, que necessita de ajustes consoante o ambiente em que se encontra. Como no conjunto de dados da validação existem uma diversidade de imagens maior, o software da empresa não se consegue adaptar a todas as imagens da mesma forma, criando assim uns resultados de métricas menos bons. Estes resultados estão apresentados na Tabela 5.1, consoante o *threshold* de área de sobreposição das unidades foliculares para a IoU.

**Table 5.1:** Comparação de *Precision* e *Recall* para diferentes valores de IOU Threshold com o melhor Treino da Rede.

Threshold IOU	Insparya		YOLOv5_OBB	
	Precision	Recall	Precision	Recall
0.3	67.12	42.57	72.0	76.1
0.4	62.9	41.0	70.7	75.4
0.5	58.27	39.15	70.9	71.4
0.6	53.99	37.35	69.6	69.7

Os resultados da validação *YOLOv5\_OBB* e do software de deteção da empresa Insparya, não apresentam resultados tão elevados como os das redes *YOLOv5* e *YOLOv8*, pois é mais difícil haver sobreposição em termos de IoU, de uma *bounding box* com orientação, do que com uma *bounding box* sem orientação. Como existem unidades foliculares que são parecidas com um segmento de reta, não existe quase largura da *bounding box* para haver uma grande sobreposição.

Como era de esperar, o *Recall* do software da empresa Insparya é bem menor que a sua precisão devido à quantidade de folículos não detetados nas imagens que correspondem aos Falsos Negativos, os parâmetros não estão devidamente ajustados para todos os ambiente no qual foi efetuada a deteção, resultando numa má deteção em algumas imagens.

Mas, através dos resultados obtidos conseguimos compreender, tanto visualmente como numericamente, que o comportamento da rede tem um grande potencial na área de transplantes capilares.

Uma das principais vantagens da rede *YOLOv5\_OBB* é a sua grande adaptabilidade a diferentes meios consoante o treino efetuado. É possível, ter cenários infinitos de possíveis ocorrências durante uma operação. A sua robustez no que diz respeito à deteção supera muito o software de deteção da empresa. Para não falar, que é possível também fazer a deteção na mesma imagem dos restantes componentes presentes no couro cabeludo do paciente, tais como, orifícios da deteção, vazamento de sangue, sujidade, a classificação das unidades foliculares, por exemplo, se tem 1,2 ou 3 folículos no mesmo orifício, entre outros.

Em relação ao tempo de deteção de ambos os métodos, a rede não precisa de um pré-processamento de imagem e nela podem ser incluídas as restantes deteções que estão a ser efetuadas à parte com o software de deteção da empresa Insparya. De facto, o tempo de deteção do software da empresa é 2.68 vezes menor, mas não é considerado uma vantagem se não for tão preciso na deteção das unidades foliculares. Estão a ser utilizados serviços em ROS para a passagem dos valores das deteções, o que pode ser um motivo também de um ligeiro atraso na passagem das deteções efetuadas e nas imagens recebidas. Isto indica que, os tempos também podem ser melhorados. Cada vez mais estão a ser criadas e atualizadas redes com uma evolução na rapidez e na precisão das deteções efetuadas, isto serve para trabalho futuro também, procurar redes mais atualizadas ou mesmo, alterar a rede em questão, de modo, a serem melhorados os tempos de computação.

Na Tabela 5.2 estão apresentadas algumas das diferentes vantagens e desvantagens de ambos

os métodos de detecção utilizados.

**Table 5.2:** Comparação de Vantagens e Desvantagens entre Rede neuronal e Métodos Tradicionais de VPC.

Tipo	YOLOv5_OBB	Software da Empresa
<b>Precisão</b>	Capacidade de aprender padrões complexos, levando a resultados mais precisos e a uma maior robustez.	Menos eficaz em lidar com algumas das variações que possam existir, levando a resultados menos precisos.
<b>Adaptação</b>	Pode ser adaptada para diferentes ambientes e situações com treino apropriado.	Geralmente requer ajustes manuais significativos para se adaptar a diferentes condições, o que é demorado e não é uma boa solução para imprevistos.
<b>Velocidade</b>	Tempos mais demorados devido ao grande uso das GPUs.	Tempos de computação mais rápidos devido apenas ao uso de cálculos básicos que permitem a detecção de unidades foliculares.
<b>Eficiência</b>	Requer um considerável poder computacional e uma grande quantidade de dados a serem no treino da rede.	Pode ser mais eficiente em termos de recursos computacionais, mas com possíveis perdas na precisão.
<b>Atualizações</b>	Pode ser atualizada e melhorada com novos dados, aumentando a sua eficácia.	Atualizações frequentemente envolvem ajustes manuais e podem ser demoradas.
<b>Escalabilidade</b>	Pode lidar com um grande volume de dados e tarefas complexas de processamento.	Pode ter limitações de escalabilidade em tarefas complexas devido a abordagens fixas.

## 5.2 Tracking Adicionado

O transplante capilar é um procedimento delicado que envolve a extração e o subsequente reposicionamento de unidades foliculares individuais de uma área doadora para uma área receptora do couro cabeludo. Cada unidade folicular é única, e sua posição e orientação corretas desempenham um papel crítico na obtenção de resultados estéticos satisfatórios. É aqui que entra o objetivo do *tracking* adicionado, que cria uma identificação específica para cada unidade folicular.

A atribuição de um identificador único a cada unidade folicular antes do procedimento permite que, os cirurgiões e técnicos de transplante capilar monitorem e registem o trajeto de cada unidade, ao longo de todo o processo. Isso, é particularmente útil quando se deseja extrair uma unidade folicular específica de acordo com as necessidades do paciente ou as diretrizes estabelecidas pelo cirurgião. O *tracking* preciso impede que ocorram erros humanos, como a perda de vista da unidade folicular desejada.

Além disso, uma unidade folicular com identificação específica oferece um registo documentado

de cada unidade folicular, permitindo que a equipa médica acompanhe o progresso do tratamento, avalie o crescimento e a taxa de sobrevivência das unidades transplantadas e, se necessário, ajuste o tratamento de forma personalizada para otimizar os resultados.

Essa adição, no contexto do transplante capilar não apenas aumenta a eficácia do procedimento, mas também melhora a segurança e a precisão, proporcionando aos pacientes uma maior confiança nos resultados desejados.

## **5.3 Trabalho Futuro**

Em relação ao futuro do projeto, existiram desafios significativos na obtenção de novas imagens para enriquecer o conjunto de dados a serem utilizados no treino. Isso deve-se, em grande parte, à pouca disponibilidade de enfermeiros para anotar imagens e à necessidade de pacientes dispostos a fornecer imagens do seu couro cabeludo durante o procedimento de transplante capilar.

### **5.3.1 Expansão do Conjunto de Dados**

A expansão do conjunto de dados para treino é uma prioridade. Aumentar a quantidade de dados disponíveis é fundamental para melhorar a precisão do modelo de deteção. Embora a obtenção de novas imagens possa ser desafiadora, encontrar maneiras de obter mais dados de qualidade é uma tarefa importante.

### **5.3.2 Adicionar Novas Classes**

Em vez de ser detetada apenas uma classe, é possível adicionar os diferentes elementos presentes no procedimento do transplante capilar, tais como unidades foliculares com 1,2 ou 3 folículos, orifícios provenientes da extração e as unidades foliculares já transplantadas. Isso pode ser uma adição que vai enriquecer a deteção em si.

### **5.3.3 Otimizar Tempos de Deteção**

Explorar maneiras de otimizar os tempos de processamento da rede e aprimorar sua integração no software da empresa representa também um dos objetivos futuros para esta investigação. Em vez do uso de serviços em ROS, tentar implementar diretamente no software da empresa.

### **5.3.4 Treinos Com Maior Poder Computacional**

Outro ponto crucial, é melhorar a capacidade computacional utilizada no treino da rede. A utilização de máquinas com GPUs mais poderosas permitirá o uso de imagens de maior resolução e um *batch size* mais significativo durante o treino, sendo essencial para melhorar a robustez e o desempenho da rede.

### **5.3.5 Explorar Novas Redes Neurais e Diferentes Técnicas de Deteção**

Explorar novas arquiteturas de redes neurais é um ponto bastante importante, pois estão constantemente a ser criadas novas redes e as existentes a sofrerem atualizações que melhoram

o seu desempenho. Como por exemplo, a rede *YOLOv8*, que introduziu métodos de detecção de *keypoints* em imagens. Esta abordagem, pode ser aplicada para identificar pontos de interesse, como os cantos das unidades foliculares. Outra possibilidade, é a segmentação de objetos, também introduzida na rede *YOLOv8*, que pode ser utilizada para criar *bounding boxes* com base nos valores dos extremos obtidos através da segmentação. Isso proporcionaria uma maneira diferente de detetar as unidades foliculares.

Um método de detecção proposto também nesta dissertação, foi a utilização de redes neurais sem orientação para garantir a existência de um folículo dentro das *bounding boxes* e a partir daí, serem procedidos os cálculos necessários para obter a orientação da unidade folicular presente dentro da *bounding box*.

# Bibliography

- [1] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, and Rob Wheeler. Ros: an open-source robot operating system. <https://www.ros.org/>, 2009.
- [2] Venus Treatments. ARTAS Robotic Hair Restoration Treatments. <https://www.venustreatments.com/en-gl/artas.htm>, 2020.
- [3] Koray Erdoğan, Onur Acun, Ayhan Küçükmanisa, Ramazan Duvar, Alp Bayramoğlu, and Oğuzhan Urhan. KeBot: An artificial intelligence based comprehensive analysis system for fue based hair transplantation. *IEEE Access*, 8:200461–200476, 2020.
- [4] Glenn Jocher. Ultralytics yolov5. <https://github.com/ultralytics/yolov5>, 2020.
- [5] Glenn Jocher, Ayush Chaurasia, and Jing Qiu. Ultralytics yolov8. <https://github.com/ultralytics/ultralytics>, 2023.
- [6] Xue Yang and Junchi Yan. On the arbitrary-oriented object detection: Classification based approaches revisited. *International Journal of Computer Vision*, 130:1–26, 05 2022.
- [7] Kai-Xuan Hsu. YOLOv5\_OBB: YOLOv5 for oriented bounding box detection. [https://github.com/hukaixuan19970627/yolov5\\_obb](https://github.com/hukaixuan19970627/yolov5_obb), 2021.
- [8] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. <https://github.com/abewley/sort>, 2016.
- [9] Patrick M. Zito and Blake S. Raggio. *Hair Transplantation*. StatPearls Publishing, Treasure Island (FL), 2022.
- [10] Ravi Sharma and Anushri Ranjan. Follicular unit extraction (fue) hair transplant: Curves ahead. *Journal of Maxillofacial and Oral Surgery*, 18(4):509–517, 2019.
- [11] Anil Dua and Kapil Dua. Follicular unit extraction hair transplant. *Journal of Cutaneous and Aesthetic Surgery*, 3(2):76–81, 2010.
- [12] John P Cole. Status of individual follicular group harvesting. In *Hair Transplant Forum Int*, volume 19, page 20, 2009.
- [13] Walter P. Unger. Hair transplantation: Current concepts and techniques. *Journal of Investigative Dermatology Symposium Proceedings*, 10(3):225–229, 2005.

- 
- [14] R. M. Rashid. Follicular unit extraction with the artas robotic hair transplant system: an evaluation of fue yield. *Dermatology Online Journal*, 20(4), 2014. Retrieved from <https://escholarship.org/uc/item/0w5620q5>.
- [15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.
- [16] Qijie Zhao, Tao Sheng, Yongtao Wang, Zhi Tang, Ying Chen, Ling Cai, and Haibin Ling. M2det: A single-shot object detector based on multi-level feature pyramid network. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):9259–9266, Jul. 2019.
- [17] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *ArXiv*, abs/2004.10934, 2020.
- [18] Mingxing Tan, Ruoming Pang, and Quoc V. Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
- [19] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12):2481–2495, 2017.
- [20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.
- [21] Eduardo Romera, José M. Álvarez, Luis M. Bergasa, and Roberto Arroyo. Erfnet: Efficient residual factorized convnet for real-time semantic segmentation. *IEEE Transactions on Intelligent Transportation Systems*, 19(1):263–272, 2018.
- [22] David W Paglieroni. Distance transforms: Properties and machine vision applications. *CVGIP: Graphical Models and Image Processing*, 54(1):56–74, 1992.
- [23] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.
- [24] Juan Du. Understanding of object detection based on cnn family and yolo. *Journal of Physics: Conference Series*, 1004(1):012029, apr 2018.
- [25] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition*, 2014.
- [26] Jasper R. R. Uijlings, Koen E. A. van de Sande, Theo Gevers, and Arnold W. M. Smeulders. Selective search for object recognition. *International Journal of Computer Vision*, 104(2):154–171, September 2013.

- [27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.
- [28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.
- [29] Ross Girshick. Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 1440–1448, USA, 2015. IEEE Computer Society.
- [30] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 06 2015.
- [31] Xue Yang, Liping Hou, Yue Zhou, Wentao Wang, and Junchi Yan. Dense label encoding for boundary discontinuity free rotation detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15814–15824, 2021.
- [32] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Edouard Duchesnay, and Gilles Louppe. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12, 01 2012.

