Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Diagnosis of ophthalmologic diseases in canines based on images using neural networks for image segmentation

Matija Buric [a], Sinisa Grozdanic [b], Marina Ivasic-Kos [a,*]

[a] *Faculty of Informatics and Digital Technologies, University of Rijeka, Centre for Artificial Intelligence University of Rijeka Ul, Radmile Matejcic 2, 51000, Rijeka, Croatia*
[b] *Animal Eye Consultants of Iowa, 698 Boyson Rd A, Hiawatha, IA, 52233, USA*

ARTICLE INFO

Keywords:
U-net
Deep convolutional neural networks
Image segmentation
Image classification
Ophthalmological diseases in dogs
Eye diseases

ABSTRACT

The primary challenge in diagnosing ocular diseases in canines based on images lies in developing an accurate and reliable machine learning method capable of effectively segmenting and diagnosing these conditions through image analysis. Addressing this challenge, the study focuses on developing and rigorously evaluating a machine learning model for diagnosing ocular diseases in canines, employing the U-Net neural network architecture as a foundational element of this investigation.

Through this extensive evaluation, the authors identified a model that exhibited good reliability, achieving prediction scores with an Intersection over Union (IoU) exceeding 80 %, as measured by the Jaccard index. The research methodology encompassed a systematic exploration of various neural network backbones (VGG, ResNet, Inception, EfficientNet) and the U-Net model, combined with an extensive model selection process and an in-depth analysis of a custom training dataset consisting of historical images of different medical symptoms and diseases in dog eyes.

The results indicate a fairly high degree of accuracy in the segmentation and diagnosis of ocular diseases in canines, demonstrating the model's effectiveness in real-world applications. In conclusion, this potentially makes a significant contribution to the field by utilizing advanced machine-learning techniques to develop image-based diagnostic routines in veterinary ophthalmology. This model's successful development and validation offer a promising new tool for veterinarians and pet owners, enhancing early disease detection and improving health outcomes for canine patients.

1. Introduction

The purpose of this research was to explore the possibility of developing efficient software models that will be able to provide fast screening for eye abnormalities in canines solely by examining an image taken by a smartphone or single camera under various conditions. Using the developed model, pet owners may have a unique opportunity to detect early development of eye diseases in their pets and seek timely veterinary help. Furthermore, such tools may also provide rapid screening capabilities for veterinarians allowing them to quickly initiate medical treatment or facilitate a referral to the veterinary ophthalmologist, dramatically improving the quality

* Corresponding author. Faculty of Informatics and Digital Technologies University of Rijeka, and Centre for Artificial Intelligence University of Rijeka, Ul. Radmile Matejcic 2, 51000, Rijeka, Croatia.

E-mail addresses: matija.buric@hep.hr (M. Buric), sgrozdan@gmail.com (S. Grozdanic), marinai@uniri.hr (M. Ivasic-Kos).

https://doi.org/10.1016/j.heliyon.2024.e38287
Received 1 April 2023; Received in revised form 4 September 2024; Accepted 20 September 2024
Available online 21 September 2024
2405-8440/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

of the veterinary care for patients with eye diseases.

To develop the precise and reliable diagnostic success of software tools, it is crucial to construct a precise and robust model based on expert knowledge [1–3]. The focus of this manuscript was the development of an image recognition routine for the common clinical symptoms observed in dogs with different eye diseases such as corneal edema, episcleral congestion, and epiphora. Furthermore, we wanted to develop a model for the recognition of the relatively frequently encountered problem in the canine population, which is a prolapse of the nictitans gland (Cherry Eye). Symbolic representation of different conditions is shown in Fig. 1.

Corneal edema Fig. 1(a) is a clinical symptom that develops as a consequence of excessive corneal stroma hydration, resulting in a bluish haze. This clinical symptom can be seen as a result of glaucoma, uveitis, corneal ulceration, trauma, corneal inflammatory disease, and corneal endothelial degeneration), and is one of the early clinical symptoms of the more serious ocular problems. The episcleral blood vessel congestion Fig. 1(b) sometimes referred to as episcleral injection is a clinical symptom that develops because of dilation and engorgement of blood episcleral blood vessels. This condition is most frequently seen as a result of glaucoma, and severe intraocular and episcleral inflammatory diseases resulting in the "red eye" appearance. This clinical symptom is frequently an indicator of the more serious ocular disease where early detection can be essential for successful treatment, and prevention of vision loss. Excessive tearing (epiphora) Fig. 1(c), is a common clinical symptom that develops as a result of excessive tear production or abnormal overflow of tears over the eyelid margin, resulting in excessive facial skin wetting. This clinical symptom is most frequently seen as a result of allergic ocular diseases but can be also seen as a result of any ocular disease resulting in excessive ocular irritation and/or pain presence. The cherry eye Fig. 1(d) is most frequently seen in young dogs because of the connective tissue weakness keeping the gland in its anatomic position below the eye globe [4]. This condition can be seen in older animals, most frequently because of the cancerous change of the nictitans gland.

As already mentioned, all the above clinical symptoms and diseases can be early clinical signs of vision-threatening ocular diseases, and their early detection and recognition may provide a unique opportunity for timely veterinary intervention and prevention of more serious complications.

The main contributions and novelties of this manuscript are the following.

a) A novel dataset of eye clinical symptoms and diseases with images containing close-ups of ocular structures, annotated and prepared for supervised machine learning.
b) Extension of U-Net architecture for image segmentation composed with the backbone of deep convolutional neural networks VGG16, ResNet34, Inception V3, and EfficientNet B3 with extensive performance comparison.
c) Influence analysis of the selected loss functions (Categorical cross-entropy loss, Dice loss, and Focal loss) in the U-Net segmentation architecture for the canine eye symptoms segmentation.
d) A novel U-Net model with Res-Net34 backbone for pixel-based automatic localization of clinical symptoms and recognition of eye diseases in dogs with detailed statistical analysis.

The rest of the manuscript is organized as follows: A summary of the research about the popular machine learning techniques used in medical-related imagery for humans and pets closely related to ophthalmology is provided in Section 2. In section 3, publicly available datasets applicable to the field of this research are introduced followed by a description of a custom dataset, specially obtained, and prepared for the recognition of canine eye diseases. Additionally, descriptions of image classifiers that are used for the U-Net backbone and transfer learning basis are summarized. In the same section, loss functions, used in the development of CNN models are described along with evaluation metrics which are used to measure the model's success. Section 4 provides a detailed description of the experiments and results. At the end of this manuscript, conclusions and further research directions are discussed.

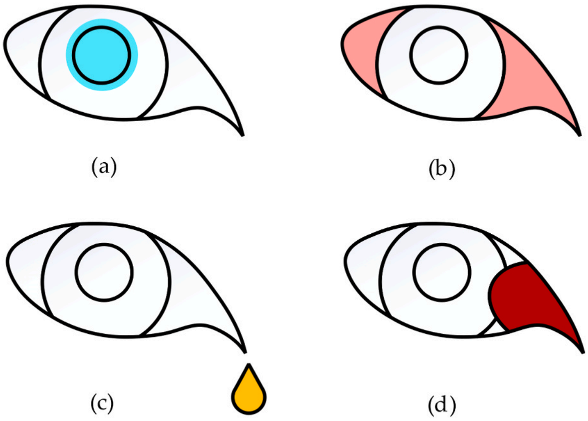


Fig. 1. Symbolic representation and general anatomic localization of clinical symptoms - corneal edema (a), episcleral congestion (b), epiphora (c), and Cherry Eye disease (d).

2. Related work

The field of ophthalmology has increasingly integrated the advances of computer vision over the past decade, leveraging its capabilities to enhance the classification and diagnosis of eye diseases [5–7]. This integration has been significantly pushed forward in recent years with the adoption of Convolutional Neural Networks (CNNs), which have brought about a new era of precision and efficiency in analyzing ocular images [8,9]. The increase of diverse datasets aimed at studying various eye conditions has been the foundation of this advancement. Notable examples include ORIGA-light [10], the RIGA dataset [11], Drishti-GS [12], among others, which have provided invaluable resources for developing and testing computational models [13–19].

Despite the broad application of these technologies in human ophthalmology, the exploration into canine ocular diseases has been relatively limited [20]. Glaucoma represents an area of canine ophthalmology that has seen some research studies, yet the availability of comprehensive datasets for such conditions remains a significant challenge [21]. The application of computer vision for ophthalmology purposes has not only advanced the diagnosis and understanding of human eye diseases but has also slowly paved the way for new diagnostic applications in veterinary ophthalmology. The application of neural networks for tasks such as eye classification, tracking, and object detection showcases the potential of these technologies to contribute significantly to the field [22–28]. Computer vision in combination with Large Language Models (LLMs) also showed promising results by providing detailed diagnoses based on the classification of symptoms [20]. Among these advancements, the use of transfer learning marks an important development, particularly in enhancing the accuracy of canine identification [29,30]. This technique exemplifies how leveraging pre-existing datasets and models can overcome challenges inherent in veterinary applications, especially in the realm of canine ophthalmology where specific conditions demand precise diagnostic capabilities. The most relatable manuscript pertinent to this research described ulcerative keratitis based on CNNs [31]. The study utilized dog eye images acquired under controlled laboratory conditions from diverse breeds, with 69 % originating from smaller breeds like Shih-Tzu, Maltese, Pekingese, and Poodle. The dataset, categorized into three classes based on corneal ulcer severity, underwent augmentation through rotation, image flipping, and a combination of both, resulting in a total of 1040 images. Classification employed fine-tuned CNN models, Inception [32], ResNet [29], and VGGNet [30] that were pre-trained on the ImageNet dataset. The findings revealed that ResNet and VGGNet consistently outperformed GoogLeNet, with the majority achieving classification accuracies surpassing 90 %.

The first convolution networks were used for classification tasks only [33]. Image classification gives a label of a certain class for an input image. This is not always the desired output because specific problems require the exact localization of the region of interest in an image. This task can generally be divided into two groups: object detection and image segmentation. Object detection takes an input image and marks the exact location of the predicted object at the bounding box level, while image segmentation, known as pixel-based classification, provides pixel-level information about the class to which the pixel belongs. For example, Fig. 2 shows the results of eye detection Fig. 2(a) and eye segmentation Fig. 2(b). Image segmentation can be further divided into semantic and instance segmentation, where semantic segmentation treats all objects of the same class as one, and instance segmentation treats all separate objects as standalone instances.

For our modeling, we first needed to detect the eye to localize the area of interest in the image, and then segment the eye image so that we can more precisely mark the area of the eye that is affected by a disease. For this reason, we chose artificial neural networks that are typically used to classify images, localize objects, and extract features from images, such as edges or corners, namely convolutional networks. These networks use convolution operations to process the input data allowing them to handle large amounts of image data and video. We also chose to research the methods that showed the best results on segmentation tasks on benchmark data sets, so in the research, we decided to investigate the possibility of applying and adapting the U-Net architecture and its variants in our tasks.

Given that there is no available data that could be used to train models for automatic canine eye disease detection, we had to build a dataset that includes segmentation of eye regions with clinical symptoms and classification into the appropriate class. To build and annotate the database, we used original images and expert knowledge from a board of certified veterinary ophthalmologists.

It is known that the training of most neural networks will be more successful if many thousands of labeled training samples are used, which is difficult to ensure when creating a customized database for special purposes such as our case. However, experiences and trends have shown that deep neural networks, including U-Net, can be successfully trained with much fewer images, using methods of knowledge transfer and data augmentation [34], so we will implement these methods in our research.

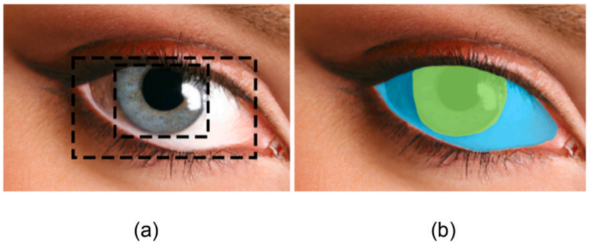


Fig. 2. Object detection (a) vs. image (instance) segmentation (b).

3. Datasets and methods used

To be able to learn a machine learning model for the detection and segmentation of the eye area, as well as for all tasks of supervised machine learning, it is necessary to have an appropriate data set prepared in such a way that the model can be trained on it and a set on which the model will be evaluated and tested. When defining an appropriate data set, it is necessary to collect appropriate images, preprocess them, and prepare them for the appropriate machine-learning tasks for which the model will be trained. In the field of medical and veterinary image analysis, we used expert annotations to associate a diagnosis label to images for the task of image classification and recognition of eye disease diagnoses, then labeling with a bounding box the object (eye) in the image and associating the corresponding label that includes the corresponding diagnosis and coloring the eye region for image segmentation. Careful selection of the learning set, i.e. a quality data set, is key to developing a model that aims to help define diagnoses with relevant and useable suggestions.

3.1. Publicly available datasets

Many publicly available datasets are designed specifically for a certain purpose, but none involve the detection or segmentation of the eye, especially not the eye of a dog or cat, nor the detection of disease states. The size of benchmark datasets varies from a couple of MB to a few TB and can be applied in classification, object detection, semantics, and instance segmentation.

The Pascal VOC dataset [35] is about 2 GB in size and consists of images labeled for object detection and segmentation. It is among the first datasets publicly available for these purposes. Objects in the Pascal VOC dataset are divided into 4 categories: people, animals, vehicles, and indoor objects. Each image has its corresponding label file which gives information about one or more objects in the image.

The MS COCO dataset [36] is larger than previously mentioned, consisting of about 40 GB of images published in 2014. under the sponsorship of Microsoft company. It provides images with information about category, localization, and semantic text description. Computer vision researchers for training and evaluating models often use it. It describes 91 classes on over 300,000 images with over 2.5 million labels. The dataset is easily obtainable and customized allowing one to search through the database over the web user interface.

The open Images dataset [37] at its current version consists of 9.2 million images with integrated annotations for image classification, object detection, and visual relationship detection. The images are under a Creative Commons Attribution license. Images are described with over 30 million labels for almost 20,000 concepts, over 15 million bounding boxes for 600 object classes, and 375k visual relationship annotations involving 57 classes. The open Images dataset was collected and organized by the Google AI team.

The ImageNet dataset [38] contains more than 14 million images and over 20,000 categories [39]. It is used as the benchmark of model performance by many computer vision researchers. The dataset is divided into ImageNet Large Scale Visual Recognition Challenges [40] of which the latest is ILSVRC 2017. It is freely available on Kaggle. ILSVRC annotations are separated into two categories: image-level annotation which gives binary results and object-level annotation in the form of a bounding box around the object

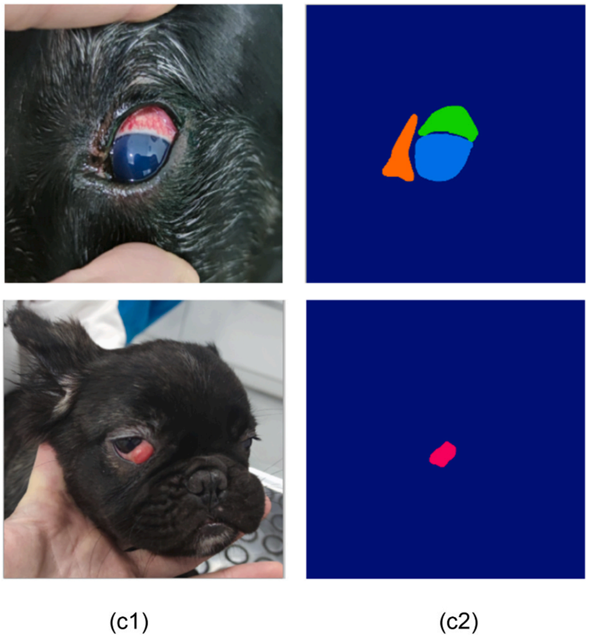


Fig. 3. The input image examples, column (c1), with the corresponding masks column (c2). The masks are colored to better distinguish between classes.

with a corresponding label.

3.2. Eye disease custom dataset

We built our original custom dataset DogEyeSeg4: Dog Eye Segmentation 4-Class Ophthalmological Disease [41] to train a model for automatic disease detection in dogs. The images we used to create the set were collected as part of routine clinical assessments at two specialist veterinary clinics by veterinary ophthalmology specialists. For the purposes of this study, images were selected at random and were anonymized, ensuring that no review dates, client information, or animal identifiers were included. There are 145 images in total.

Some of the images were taken from the Atlas of Veterinary Eye Diseases [4]. All images were scanned by an expert, a specialist in veterinary ophthalmology. As the images were collected in real conditions, in the wild, in some images the dog's eye is in the foreground, while others include the entire head and even the entire body. The recording was done with a smartphone or camera from different recording positions and distances from the object, in different resolutions and lighting conditions, so the task of recognizing and segmenting eye diseases in the images is very challenging. All images are transformed to a size of 320x320 px corresponding to the network input.

For each image, a corresponding single-channel mask of the same dimensions as the original image is created that includes segmented image regions corresponding to a particular medical symptom or eye disease. There are 4 classes plus the background in the set, which are displayed in such a way that different pixel intensities represent different classes as shown in Fig. 3. Column Fig. 3(c1) describes original images and Column Fig. 3(c2) corresponding masks showing ground truth.

The dataset exhibits a well-balanced distribution among the different classes, as can be observed in Fig. 4.

3.2.1. Augmentation

To enhance the robustness and effectiveness of our model, we implemented data augmentation techniques. Drawing from the insights of previous research in the domain of medical imaging, particularly referencing the U-Net approach [42], we sought to expand our dataset without duplicating images or altering the inherent symptoms. The fundamental objective was to simulate diverse camera positions during the acquisition of images, thereby introducing variability without modifying the underlying pathological characteristics. Researchers studying medical images, akin to our approach, observed that certain augmentations could positively impact model performance. For instance, applying a horizontal flip to input images demonstrated noticeable enhancements when the number of cases approached 200, albeit with diminishing returns beyond this point.

In a separate context, those exploring polyp segmentation [43] have noticed that changes in brightness and contrast have a favorable influence on one of their models, while another model benefited more from rotation and shear transformations. We have decided to use the same principle to create more input data as presented in Fig. 5. Therefore, we have applied affine transformation to original image Fig. 5(a), such as horizontal flip Fig. 5(b), horizontal shift Fig. 5(c), rotation of±15° Fig. 5(d), and translation of 50px Fig. 5(e) and we formed a dataset of 200 images. It is worth mentioning that the zoom function wasn't used since it adds interpolation and that clutters masks with values other than initially set up.

The bottom line of our augmentation methodology lies in its ability to diversify the dataset, making it more representative of different viewing angles during image capture, all the while preserving the integrity of the underlying symptoms. This approach aligns with a scientific and rigorous manner of data manipulation, ensuring that the augmentations serve as a surrogate for changes in camera position, without altering the essential clinical characteristics of the eye conditions under study.

3.3. Deep convolutional neural network models and transfer learning

Transfer learning is a widely used technique in computer vision that allows the use of patterns previously learned on other problems and data sets with little tweaking, on another data set to learn models of interest instead of starting the learning process from scratch. Namely, CNNs in their layers can learn different features on images, e.g.: initial network layers learn low-level features such as lines,

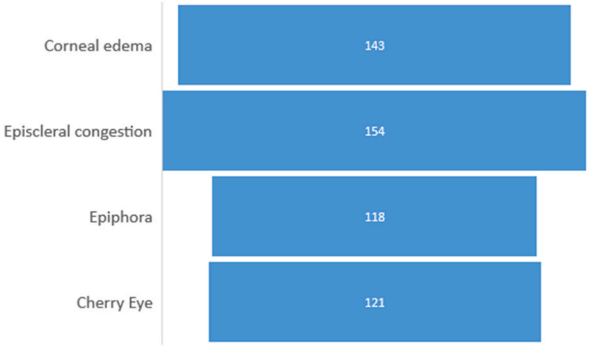


Fig. 4. Distribution of Data Classes in the Dataset. The number represents the recurrence of certain diseases in images.

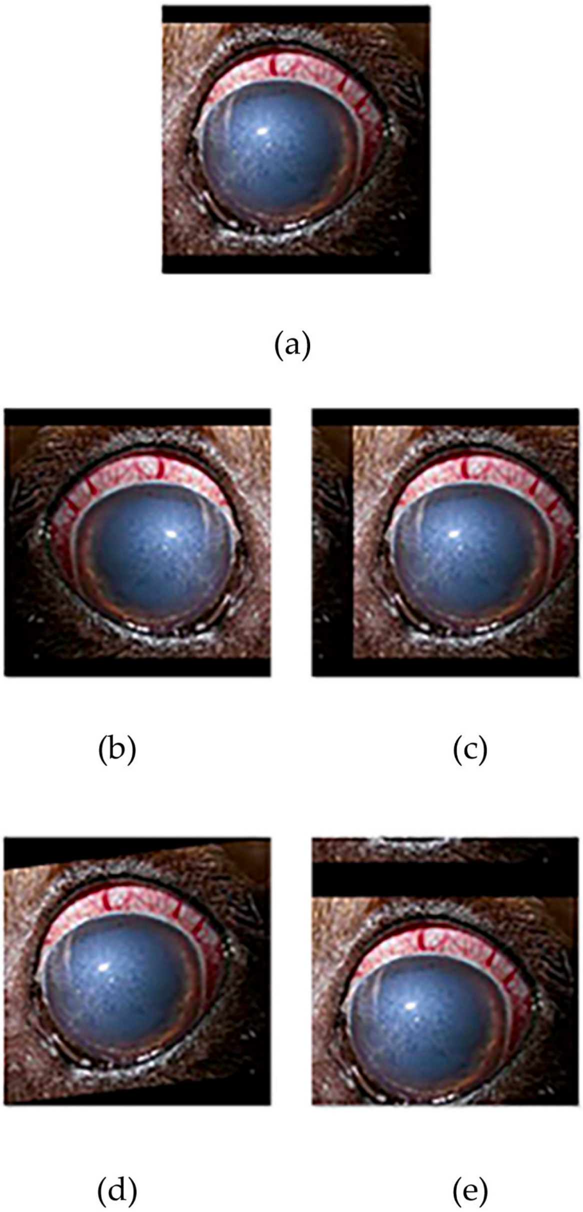


Fig. 5. Affine augmentations: Original image (a), horizontal flip (b), horizontal shift (c), rotation (d), and vertical shift (e).

points, curves, and the like, layers in the middle of the network learn objects built on top of low-level features while high-level layers can include high-level features based on the features of previous layers. Of course, the more similar the data sets are, the better results can be expected. It has been shown transfer learning achieves as good results as learning from scratch but with significantly less time and computer resources [44].

Also, thanks to transfer learning more complicated CNN architectures can solve more complex computer vision tasks, so we can build an architecture e.g. object detection or image segmentation on top of another CNN that was originally trained for image classification, we just need to discard the fully connected layers that are intended for classification. In this case, we use CNN as a feature extractor, which is a key part of the model for object detection and segmentation. The networks that are used to extract features in more complicated models and that process the input data into a representation of a particular feature are called the backbone. Many popular CNN architectures can be used as backbone networks, and we have included in our research: VGG (Visual Geometry Group Very Deep Convolutional Network) [30], ResNet (Residual Neural Network) [29], Inception [9,10], EfficientNet [45].

Those networks that we used to extract features work well as stand-alone networks on simpler classification tasks, and we have used their already pre-trained models on large data sets such as the ImageNet dataset [40] to initially adjust the network parameters to their pre-trained weights.

The pre-trained models, initially developed on the extensive and varied datasets serve as a strong foundation for their adaptability to specific tasks on the custom ophthalmology dataset. This custom dataset is carefully constructed to be used in eye disease detection; tailoring domain-specific features built on the rich knowledge gained from ImageNet. Through a process of fine-tuning, these models were trained to provide the best possible results in the specific area of ophthalmic image segmentation for our research. This fine-tuning involved adjusting model parameters and training strategies to match the unique characteristics and challenges presented by ophthalmic images. As a result, the models are finely calibrated to detect and diagnose eye diseases with high accuracy. To incorporate pre-trained model weights into U-Net architecture it was essential to prepare images to an expected input of the desired model. The desired model is applied to a coder part of the U-Net structure and feature maps are concatenated in the same manner as when the U-Net base model is applied to a decoder segment. This way U-Net becomes the structure that is held by a CNN backbone of one's choice.

Further, different backbones used in these experiments will be described along with the advantages and disadvantages of each one.

The fully convolutional neural network VGG, which was proposed in 2015, was designed to reduce the number of parameters in the convolution layers and thus reduce the training time. Compared to AlexNet [46], which was the network of choice at the time of publication, VGG is much simpler but highly efficient. VGG is very similar to the U-Net base model, consisting of convolution layers and ReLu activation functions where the early layers take care of low-level shape recognitions like lines and edges while deeper levels take care of more complex shapes. After each convolution layer/ReLu block comes the max pooling layer which finally ends with the softmax layer for image classification. It's worth mentioning that, compared to original VGG which fundamental role is classification, dimension in the output layer of backbone is adjusted to several segmentation classes. It comes in two models: VGG16 and VGG19 where numbers 16 and 19 determine the number of layers. The preliminary results in our research show better performance and accuracy of VGG16 over VGG19 which corresponds to research [47] that suggests it will result in even worse performance with a rising number of classes [47]. Therefore, the VGG16 architecture as shown in Fig. 6 will be used as one of the backbones in this research. The main advantage of VGG is popular today is that it provides researchers with a state-of-the-art model that was trained for 2–3 weeks on high-end GPUs freely.

With each layer in CNN, the complexity of model functions increases. The weights of CNN, such as VGG are computed through the backpropagation. Due to backpropagation the gradient loss closely related to the weights is calculated in each layer and reverted through the network. With each multiplication or convolution of small weights, the gradients get considerably smaller in earlier layers, which can result in a significant increase in the training time. This is known as the vanishing gradient problem and considerably affects the performance of CNN [48]. To avoid this the number of layers should be reduced or another way of keeping the gradient significant should be designed. One of the designs to address the vanishing gradient is described in ResNet CNN. The ResNet keeps gradient value 1 through its identity function. A ResNet uses shortcut connections, which makes conventional CNN a residual neural network. By doing so it allows gradient to pass via those connections and prevents them from diminishing while lowering the complexity by use of fewer filters. This can be observed in Fig. 7 where the simplest ResNet is described. As with VGG, two or more-digit numbers beside ResNet imply how many layers the network has (ResNet18, ResNet34, ResNet50, ResNet101, etc.). For this research, ResNet34 was hand-picked following the guidelines in Ref. [47].

In certain cases, objects in focus appear in different sizes and this is proven to be difficult to detect, especially if the dataset doesn't consist of different scale object presentations [49]. In one of the well-known publicly available datasets MS COCO [36] many object instances occupy from 1 % to 50 % of the original image area [49]. The CNN which addresses this issue was released by the authors of [32] under the name Inception. The Inception CNN brings a new type of repeating CNN block called Inception models. These Inception models hold different-sized convolution layers: 1x1, 3x3, and 5x5 each designed for the extraction of features at different object sizes.

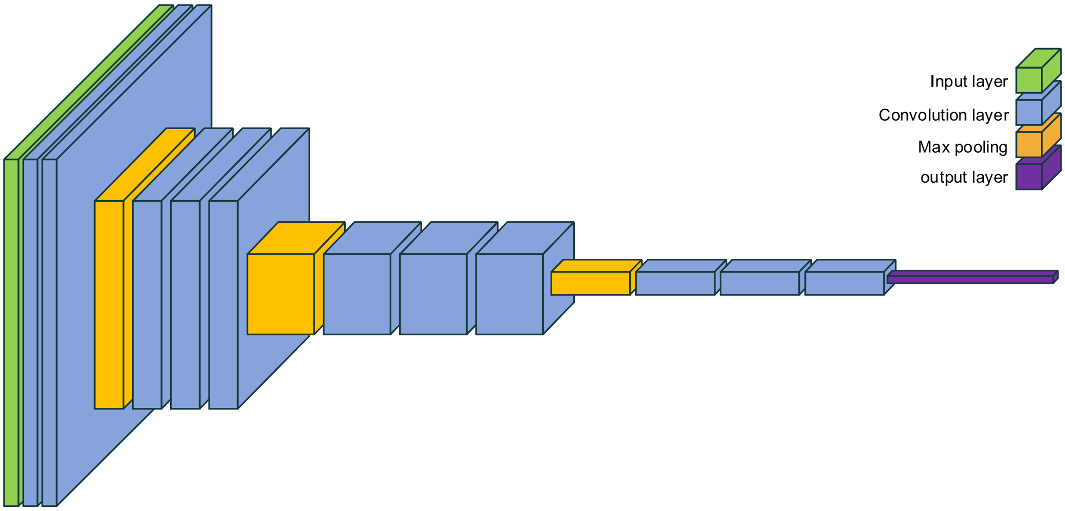


Fig. 6. The architecture of a VGG-16 deep convolutional neural network.

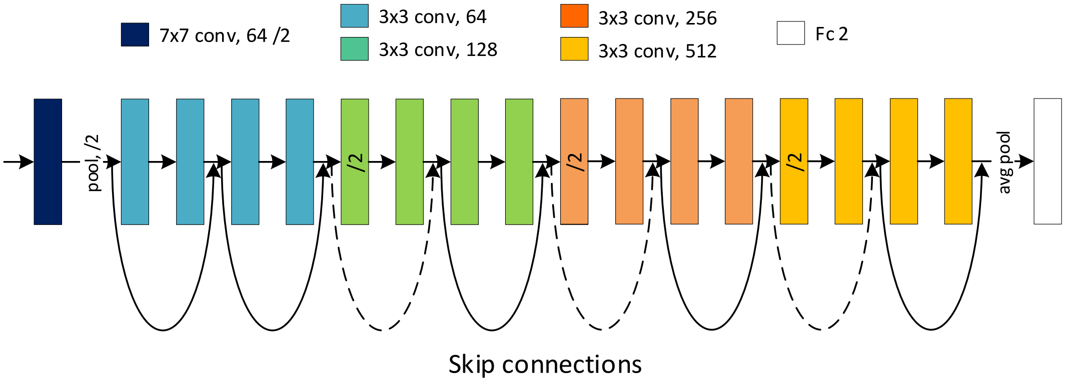


Fig. 7. The architecture of the ResNet18 deep convolutional neural network.

By combining them in parallel Inception models carry more relevant information but at a higher computation cost. Inside an Inception module padding to the max-pooling layer is applied to keep the 2D dimensions of input size the same as the output. Because of that concatenation was made possible where height and width don't change but only the number of channels. The architecture of Inception V3, which was used in this research is shown in Fig. 8.

The last CNN used in this research, called EfficientNet, tries to efficiently use the knowledge gathered from the previously introduced CNN-s. The architecture of the net is carefully designed to give better results with fewer parameters. As an example, EfficientNet B1 is 7.6x smaller and 5.7x faster than ResNet152 [45]. The factors that were dealt with to achieve this are depth, the width of the net, and the resolution of input images. A deeper network means a greater number of layers that capture more complex features, but a vanishing gradient makes it hard to train. Wider networks are easier to train, giving more detailed features but saturate very rapidly. Higher resolution input images will provide more detailed features when trained but to a certain point after it diminishes. The authors of EfficientNet CNN began with a smaller base model and by combining all three mentioned factors simultaneously started scaling the network until the desired results were met. This method was called compound scaling. In our research EfficientNet B3 has been proven to be the best ratio of precision and speed, therefore future experiments were commenced using it. In Fig. 9., the architecture of EfficientNet base model is presented.

The pre-trained model depends on the usability of the dataset used for training.

3.3.1. *U-net base model for image segmentation*

U-Net is an image segmentation architecture that was developed primarily for medical image analysis [5]. It takes an input image and predicts a segmentation mask for each pixel in the image, indicating the class of the pixel (e.g., disease, background). The U-Net architecture can be divided into two parts: an encoder and a decoder.

The encoder is a convolutional neural network (CNN) that downsamples the input image and extracts features at different levels of abstraction. It consists of a series of convolutional and pooling layers. The base U-Net [50] CNN has a kernel size of 3x3 and uses the ReLU activation function for each convolutional layer and each pooling layer is a 2x2 maxpooling layer. The encoder down samples the input image by a factor of 2 at each stage. This means that the output of the encoder has half the spatial resolution of the input image.

The bridge connects the encoder and decoder. It combines the features from the deepest layer of the encoder with the features from the shallowest layer of the decoder. By doing so it allows the network to propagate contextual information from the encoder to the decoder. This is essential for accurate segmentation.

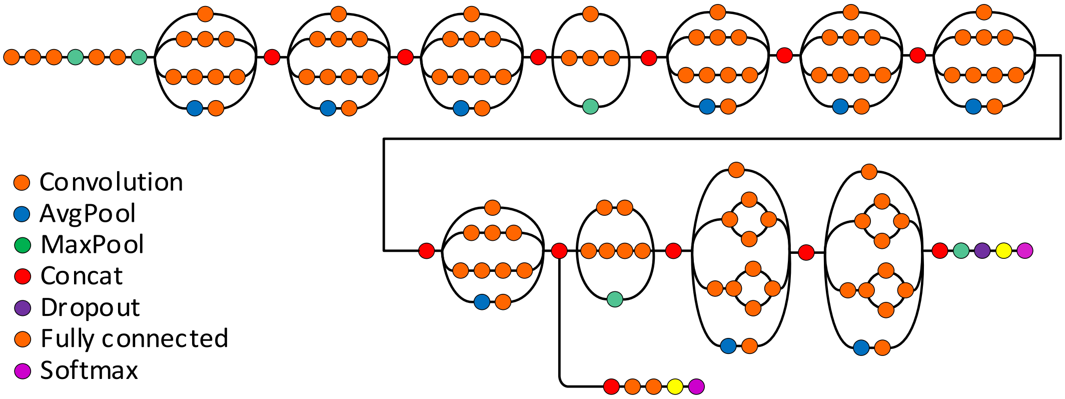


Fig. 8. The architecture of Inception V3 deep convolutional neural network.

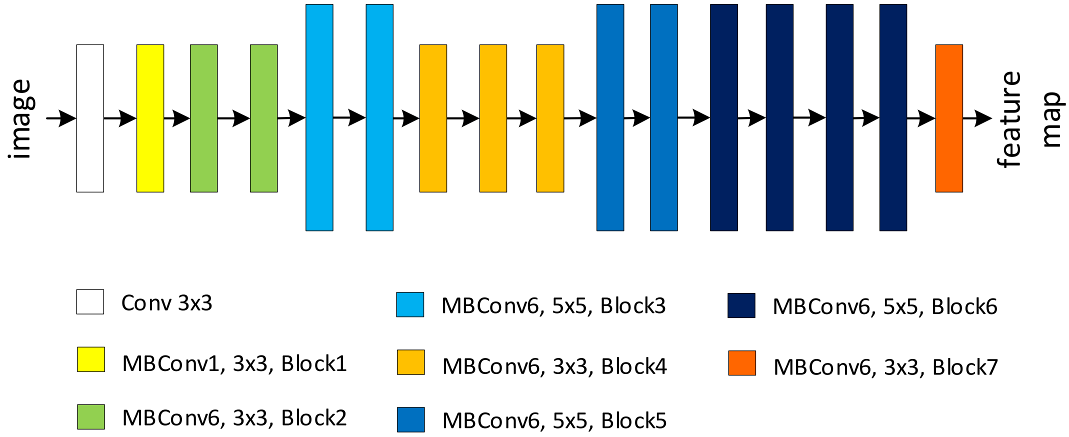


Fig. 9. The architecture of EfficientNet B0 deep convolutional neural network.

The decoder is also a CNN, but it up-samples the feature maps from the encoder and combines them with features from the encoder at different levels of abstraction. The decoder consists of a series of up sampling and convolutional layers.

As before the base U-Net CNN has a 2x2 transpose convolution layer in each up-sampling layer and a kernel size of 3x3 which uses the ReLU activation function.

The decoder up-samples the feature maps from the encoder by a factor of 2 at each stage. This means that the final output of the decoder has the same spatial resolution as the input image.

The following algorithm diagram Algorithm 1. explains the steps involved in training a base U-Net model. The Input is a dataset of training images and ground truth segmentation masks, and the output is a trained U-Net model. The weights of the encoder and decoder networks are initially randomized. When pre-trained models are employed, the weights are typically initially set based on the learned parameters from the pretraining process. For each training image and its corresponding ground truth segmentation mask, the input image is encoded using the encoder network, followed by the concatenation of the encoder feature maps with the corresponding bridge feature maps. The concatenated feature maps are then decoded using the decoder network to generate the predicted

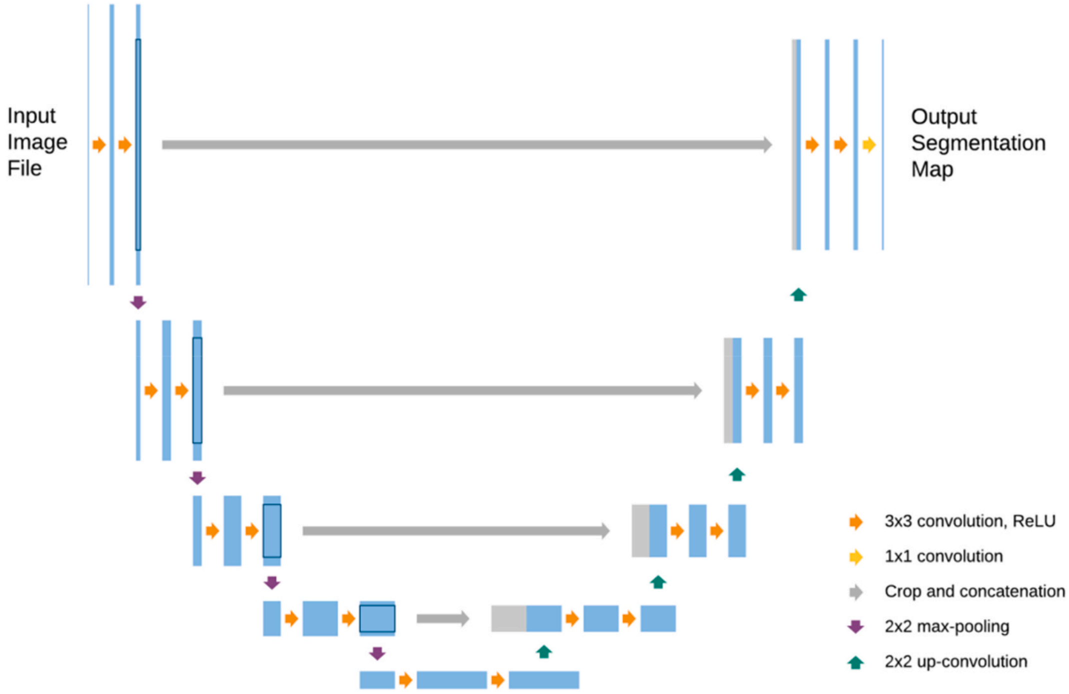


Fig. 10. Basic U-Net architecture. The arrows represent convolution, ReLu, max-pooling, and crop-concatenation. Blue boxes feature maps at each layer and gray cropped feature maps from the contracting path. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

segmentation mask. The dissimilarity between the predicted segmentation mask and the ground truth segmentation mask is calculated to derive the loss. The weights of the encoder and decoder networks are subsequently updated to minimize this loss. This process is iteratively repeated for each training image in the dataset until convergence, signifying the point at which the model achieves a minimum or satisfactory loss level.

Algorithm 1
Algorithm explaining the process of getting predicted segmentation masks from training images with corresponding masks describing ground truth.

```
ALGORITHM 1: U-NET ALGORITHM
Input: training image and ground truth segmentation mask
Output: trained U-Net model
Initialize encoder and decoder network weights randomly
Training Loop:
    For each training image and ground truth segmentation mask:
        Encode image using encoder network
        Concatenate encoder feature maps with bridge feature maps
        Decode concatenated feature maps using a decoder network to generate
        predicted segmentation mask
        Calculate loss between predicted and ground truth segmentation masks
        Update encoder and decoder network weights using an optimization
        algorithm
Convergence Check:
    If loss converges or reaches a satisfactory level:
        Exit training loop
    Else:
        Continue training loop
```

Once the model is trained, it can be used to segment new images by predicting the probability of each pixel belonging to each class. The class with the highest probability is then assigned to each pixel. A visual representation of the algorithm can be seen in Fig. 10 in the form of a U-shaped letter which propagates information along the architecture allowing it to segment objects in a particular area using relevant information from a larger overlapping area.

The upgraded U-Net model utilized in this study combines the strengths of different backbones, employing a feature extraction at various abstraction levels, like the structure depicted in Fig. 11. Importantly, these backbones leverage transfer learning, being pre-trained on ImageNet datasets. This ensures the incorporation of features previously fine-tuned on a diverse range of images, enhancing the model's capacity to recognize and represent sophisticated patterns in the specific dataset under investigation.

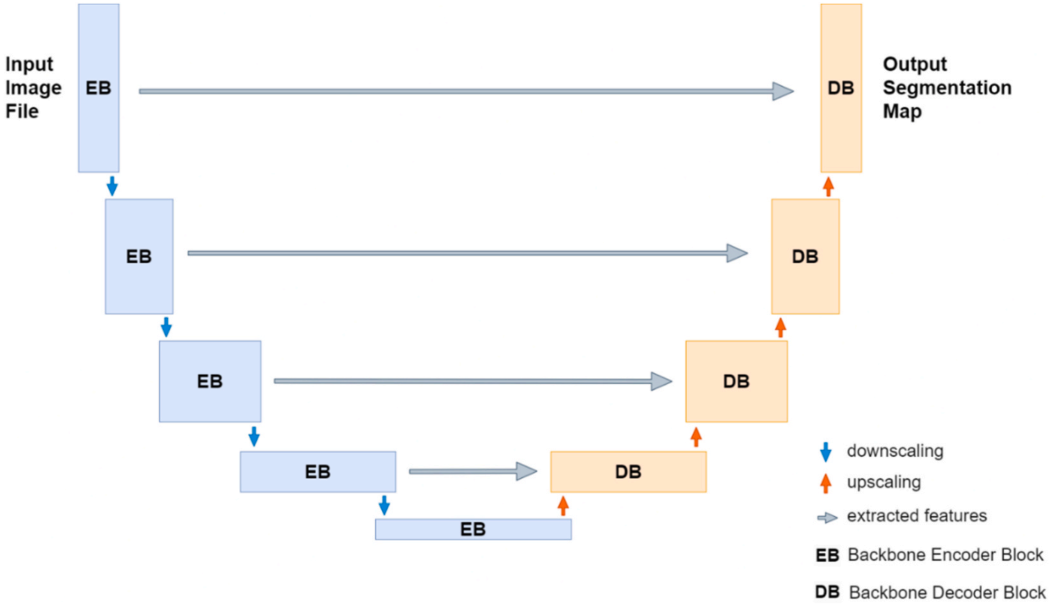


Fig. 11. Improved base U-Net model with backbone where each encoder block downscales input image and extract features which are then combined with decoder blocks up-sampled feature maps at different levels of abstraction.

3.3.2. Activation function

The purpose of the activation function in CNN-s is to calculate and convert the weighted sum of the neuron to a non-linear output. The most used activation functions and the ones used in the scope of this research are ReLu (Rectified Linear Unit) [51], Sigmoid [52], and SoftMax [53]. Their graphical presentation is shown in Fig. 12 where Fig. 12(a) describes ReLu, Fig. 12(b) Sigmoid and Fig. 12(c) SoftMax.

ReLu is often used in hidden layers of CNN. **Equation (1)** of the ReLu function is:

$$ReLu(x) = \left\{ \begin{array}{c} 1\ x > 0 \\ 0\ x \leq 0 \end{array} \right\}, \tag{1}$$

It avoids vanishing gradient, and it is simpler than sigmoid activation functions and therefore it requires less computation resources. Its main disadvantage is shown to be for the activations in the part of the domain less than 0 causing dead neurons. Some derivatives avoid such behavior, such as Leaky and Parametric ReLu but cause other problems [54].

The sigmoid activation function has an output from 0 to 1 and because it is not zero-centered it raises the problem of vanishing gradient which results in great computation time consumption. **Equation (2)** of the Sigmoid function is as follows:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}}, \tag{2}$$

Its main advantages are simplicity and applicability to be used as a classifier. In image segmentation, it is widely used for binary classification. For the multiple classification SoftMax activation function is used rather than Sigmoid. SoftMax as shown in **Equation (3)**.

$$SoftMax(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}, \tag{3}$$

When activation functions are applied for multiple classifications, SoftMax's probabilities sum to a value of 1 which is not the case with a Sigmoid's probabilities. Sigmoid's class probabilities are independent of one another because Sigmoid treats each neural node output value separately. In SoftMax, where probabilities depend on each other, one probability value will decrease in favor of a more probable class. SoftMax for this reason is easier to implement as it gives a vector as an output from which a class can be easily extracted using Argmax.

3.4. Evaluation

The evaluation metric is used to measure the quality of the trained model to determine its usefulness. There are many different evaluation metrics available to assess a model. Choosing the right one has a great impact on the training and testing and therefore on acceptance of the same. To justly compare models, the same metric must be used. Some metrics are more applicable to image classification than to image segmentation and vice versa. Precision is one of those. It is the simplest and most effective metric but due to class imbalance, it is shown to be extremely inefficient. The Jaccard index, also known as Intersection over Union (IoU) and Dice

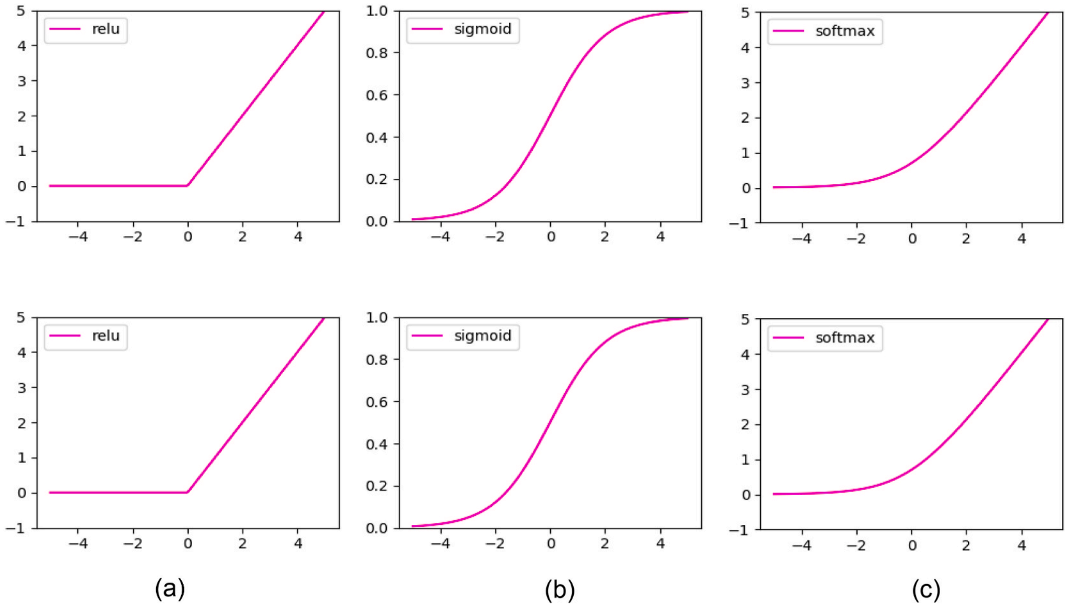


Fig. 12. Activation functions from left to right: ReLu (a), Sigmoid (b), SoftMax (c).

similarity coefficient (DSC) are the most used in region detection metrics.

In image segmentation the IoU, as the name suggests, takes two regions of interest, and measures the similarity between them by dividing the region which overlaps with a region of both regions united. If there are no intersecting pixels the IoU value will be zero but if there are some the IoU score will rise to the value of 1 in case regions overlap one another entirely. The IoU metric is simple and easy to implement once the regions are classified as shown in Fig. 13. The arrangement of Ground Truth and Prediction pixels, as shown in the same figure, is known as the binary confusion matrix and it is often used in image prediction evaluation for various metrics. Using the matrix, the DSC can be calculated, resulting in the same distribution as IoU.

The way IoU and DSC are calculated is very similar where their score is always within a factor of 2 as seen in **Equation (4)**:

$$\frac{DSC}{2} \le IoU \le DSC, \tag{4}$$

They are also always positively correlated, meaning that if one of classifiers is better under one metric it will also be better using another metric as well. The difference between metrics is visible when the average score over a set of inferences is taken. This can be observed by quantifying how much worse one classifier is over another. IoU leans towards punishing bad classifiers' single instances more than DSC does. Consequently, DSC leans toward measuring a bad classifier closer to average whereas the IoU tends to measure the score of a bad classifier closer to the extreme. This can have an impact on the final decision like in an example where most of the inferences are slightly better by one classifier and just a small number are drastically worse by the same classifier. In that case, those two metrics may differ in judgment on which classifier to prefer. In this research average scores of both metrics were used to prevent misclassification.

3.4.1. Loss function

Loss functions are mathematical formulations that analyze deviations of the predictions and the ground truth values. A greater loss value indicates the model is more prone to errors whereas a lesser loss suggests the model's predictions are more accurate. The main objective of loss function application is loss reduction. The way this is achieved is by affecting trainable parameters, for instance, biases and weights.

There are many loss functions available today and each of them has its ways of rewarding and penalizing the model's accuracy and errors respectively. Some are more applicable to the model on which this research is focused than others, which is why it is important to choose the one that will provide the best results.

One of the loss functions used in this research is the Dice loss (DL) function. It is based on the DSC metric, which essentially measures the overlap between two sample areas of an image. The mathematical **Equation (5)** is similar to DSC where DSC is subtracted from 1 to make it decrease.

$$LS = 1 - DSC, \tag{5}$$

DL comes in a range from 0 to 1, where DL leaning towards 1 indicates bad classifier and DL leaning towards 0, which is the desirable outcome, points to a perfect classifier. DSC can be expressed using precision telling how accurate positive predictions are and recall which describes coverage of actual positive samples. Both precision and recall in a good classifier will lean towards 1. The mathematical presentation of precision and recall, as seen in **Equation (6),** in terms of Type I and Type II errors are as follows:

$$precision = \frac{TP}{TP + FP}, recall = \frac{TP}{TP + FN}, \tag{6}$$

Which describes the DSC as the same as the F1-score, **Equation (7)**:

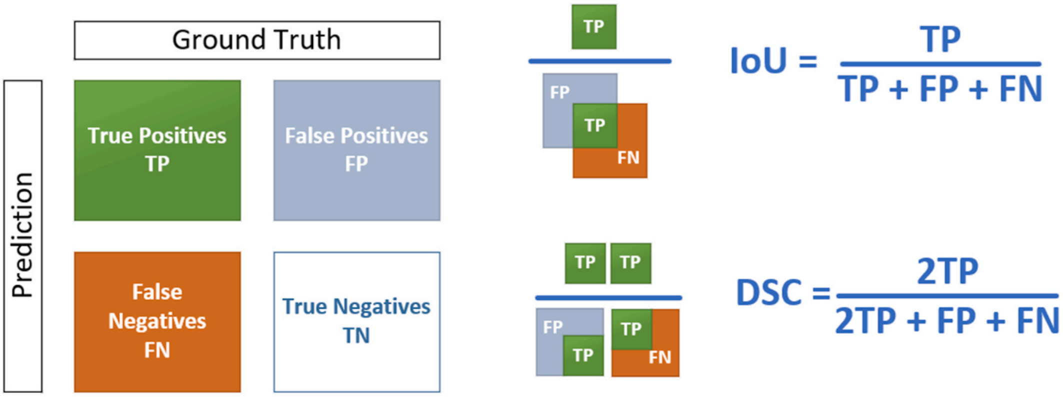


Fig. 13. Implementation of Jaccard index (IoU) and Dice Similarity Coefficient (DSC) using a confusion matrix.

$$F1 = \frac{1}{\frac{1}{precision} + \frac{1}{recall}} = \frac{2 \cdot precision \cdot recall}{precision + recall}, \tag{7}$$

F1-score is a harmonic means of precision and recall. The harmonic mean will always give the lowest value of precision and recall and by doing so it will penalize the lowest score. By maximizing F1-score both precision and recall will raise in value while staying balanced.

To transform the discrete function of DL to calculate probabilities in the range from 0 to 1, Equation (8) is used:

$$DL(y, \widehat{y}) = 1 - \frac{2\sum y \cdot \widehat{y}}{\sum y + \widehat{y}}, \tag{8}$$

where y and $\widehat{y}$ stand for ground truth and predictions of the input dataset. This way DL is easily calculated and vectorizable and generally works better when is applied on whole images than on individual pixels.

Cross entropy (CE) loss function is another loss function used in this research and it's widely acceptable in the scientific community for its simplicity and positive training results. It measures the difference between two probability distributions rather than overlaying between areas like DL for a specified arbitrary set of events. When used in statistics, entropy refers to the measurement of disorder providing info about the number of ways a system can be arranged. A higher entropy value means higher disorder. CE stands for the sum of all entropies of all the probability predictions. The CE is also known as a Binary CE (BCE) since it's used in the classification of only two classes. Equation (9) used to calculate BCE is the following:

$$BCE(y, \widehat{y}) = -(y \cdot \log(\widehat{y}) + (1 - y) \cdot \log(1 - \widehat{y})), \tag{9}$$

A Categorical CE (CCE) that consists of multiple BCEs is used in multi-class cases, and the Equation (10) for CEE is:

$$CCE(y, \widehat{y}) = -\frac{1}{\mathrm{N}} \sum_{i=1}^{N} \sum_{c=1}^{C} y_{i,c} \log\left(\widehat{y}_{i,c}\right), \tag{10}$$

where $y_{i,c}$ represent ground truth labels and $\widehat{y}_{i,c}$ a matrix of predictions for each class. The *c* is used to iterate through all classes and *i* is used to iterate over all pixels.

CE loss is focused on minimizing pixel-related error, but in class-imbalanced cases, this leads to over concertation on larger objects, which as a result, lowers the segmentation performance of smaller objects.

To improve performance on hard-to-detect objects CCE can be extended to Focal loss function (FL). FL concentrates more on the objects that are wrongly predicted than on the objects that were correctly predicted with high confidence by making sure that predictions on hard examples get better in time. This is achieved by downweighing as described by authors in Ref. [55]. If the CE Equation (11) is simplified and rewritten as

$$CE(y, \widehat{y}) = \left\{ \begin{array}{c} -\log(\widehat{y}), y = 1 \\ -\log(1 - \widehat{y}), y = 0 \end{array} \right\}, \tag{11}$$

The BCE can be presented in Equation (12):

$$BCE(y, \widehat{y}) = -\log(\widehat{y}), \tag{12}$$

at this point modulation factor is applied to the BCE, Equation (13):

$$FL(y, \widehat{y}) = \alpha(1 - \widehat{y})^{\gamma} \cdot BCE(y, \widehat{y}), \tag{13}$$

where α parameter of FL controls the class weights and γ parameter, also called as focusing parameter, controls degree of Down Weighting for hard-to-classify pixels. If the focusing parameter is equal to zero, Focal loss behaves as BCE. In the case of multi-class classification BCE is replaced with CCE, α is replaced with vector of class weights, and the $\widehat{y}$ is presented as a matrix of probabilities for each class.

4. Results

4.1. Environment

For building models, along the U-Net base model, U-Net with four different backbones was used.

- ResNet34;
- Inception V3;
- VGG16;
- EfficientNet B3.

Selection was based on popularity and benefits explained in the previous section along with publicly available pre-trained weights for quicker and better convergence. Particular versions of each backbone were hand-picked after initial experiments considering the speed-accuracy ratio.

A learning rate of 0.0001 with a fixed batch size of 16 images, optimized with Adam optimizer for 100 Epoch was used across all models.

Four different loss functions were used.

- Categorical cross-entropy loss – CCE,
- Dice loss – DL,
- Focal loss – FL,
- Balanced Dice and Focal loss function - DL + FL.

The training was done using the original dataset in combination with augmented images. The dataset was divided between test and training in a ratio of 1:9 by random selection of images. All models were developed in Python with libraries compatible with Keras and TensorFlow frameworks.

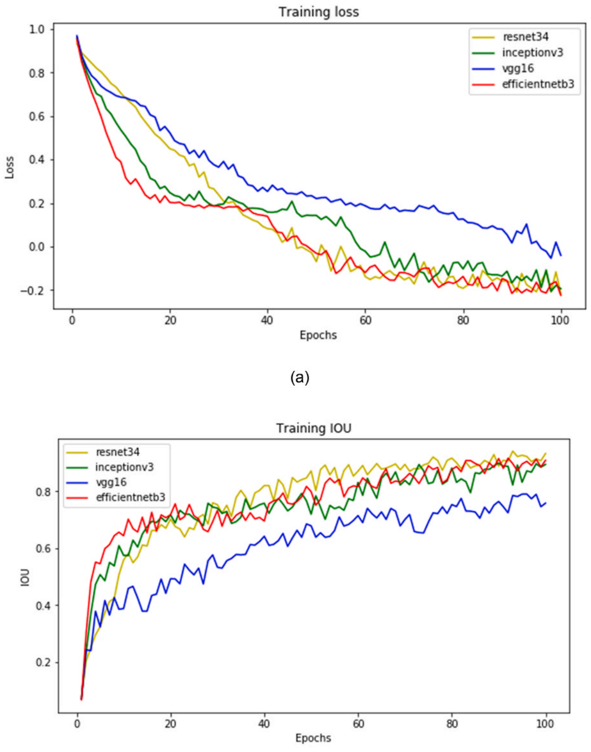
The training part was performed on the Windows platform using NVIDIA GeForce RTX 2060 GPU with 12 GB RAM and the testing part was made using Intel(R) Xeon(R) CPU E5-1603 v4 @ 2.80 GHz.

4.2. *Experiment results*

Models were trained in sequence on the same hardware using the same datasets to maintain the fairness of evaluation. During training, loss, and IoU values were observed. Metrics used to evaluate the performance of models are IoU and F-score equally distributed with a threshold of 50 %. The models with the highest score during each of the 100 epochs were chosen and preserved for further evaluation. The progress of training loss is shown in Fig. 14(a) and training IOU in Fig. 14(b).

Trained models can be divided into U-Net base model and 4 groups where each group is trained using based U-Net and with a different backbone. Each group can be subdivided into 4 different models based on loss function as can be seen in Table 1.

The results clearly show that the use of a more complex model that includes U-Net and backbone models was a very good strategy that contributed in most cases to a significant improvement in segmentation and classification results. Regardless of which type of backbone network is used and which loss function is used in model training, significantly better results than the original U-net network were achieved for all classes except for the Cherry eye class. This can be related to the insufficient data for that class which was not



(a)

(b)

Fig. 14. The example of average training loss (a) and IOU (b) values for 100 epochs.

Table 1
Segmentation results of U-Net model and composition of U-Net model with 4 different backbones of are given in Jaccard coefficients. A higher value is a better result. Four variations of each model were examined based on the loss function used in training (CCE, DL, FL, DL + FL).

Architecture	Loss func.	Corneal edema	Episcleral Congestion	Epiphora	Cherry Eye	mean Jaccard index	avg. train time[a]	avg. test time[b]
U-Net	CCE	38.7	37.5	0	47.9	31	4.5h	0.5s
	DL	36.2	35.1	0.1	58.9	32.6		
	FL	41.1	37.7	0	56.8	33.9		
	DL + FL	38.5	44	0.3	55.8	34.7		
U-Net + ResNet34	CCE	62.3	63.9	20	37.2	45.9	6h	1s
	DL	70.8	62.1	**64.6**	32.8	57.6		
	FL	68.5	78.2	59.9	53.2	65		
	DL + FL	73.9	**80.6**	38	**73.9**	**66.6**		
U-Net + Inception V3	CCE	64.1	62.9	0.9	36.2	41	8h	2s
	DL	71.2	68	31.6	61.1	58		
	FL	62.1	65.6	18.2	21.4	41.8		
	DL + FL	**78.3**	78.3	37.9	54.2	62.2		
U-Net + VGG16	CCE	63.1	66.6	1.3	33.8	41.2	6h	0.5s
	DL	67.1	60.4	31.4	27.4	46.6		
	FL	66.1	63.8	29.3	41	50.1		
	DL + FL	75.1	75.7	27.2	54.1	58		
U-Net + EfficientNet B3	CCE	66.4	64.5	2.9	37	42.7	7h	3s
	DL	67.6	63.7	31.9	31.9	48.8		
	FL	67.3	70.1	28.7	38	51		
	DL + FL	69.7	79.2	36.1	67.5	63.1		

[a] Average test time is computed as approximation of time needed for the segmentation of 20 images on Intel(R) Xeon(R) CPU E5-1603 v4 @ 2.80 GHz.

[b] Average train time is time needed for training on full dataset using NVIDIA GeForce RTX 2060 GPU with 12 GB RAM.

sufficient to adjust the EfficientNet network parameter, and the overly simple VGG backbone network which failed to extract relevant and discriminative features for Cherry eye.

The U-Net + ResNet34 model using DL + FL loss has the overall highest score for two out of 4 classes according to the Jaccard coefficient measure. When exclusively employing the Dice loss within the U-Net + ResNet34 model, performance, as measured by the Jaccard coefficient, demonstrates a significant improvement of the Epiphora class compared to other models. U-Net + Inception V3 has the best score for the Corneal Edema class with a Jaccard index of 78.3 %. The difference between the best and worst performing models in cases where the background class is disregarded is 27 %. Both are from the model with the same backbone. This is also an example of the highest improvement using different loss functions. If the least detectable class is not considered – Epiphora class, the average score of all models improves up to over 13 %. The models based on U-Net + VGG16 are shown to be the quickest in the segmentation of test images.

4.2.1. Evaluation of models using 5-fold cross-validation

Further experiments were performed on all models using DL + FL functions which gave the best results according to Table 1. Subsequently, since our custom set is in relative modes we wanted to use all the data we have available for training and testing, so we did use a k-fold cross-validation where k is set to 5. The dataset is divided into several folds or subsets; one of these folds is used as a

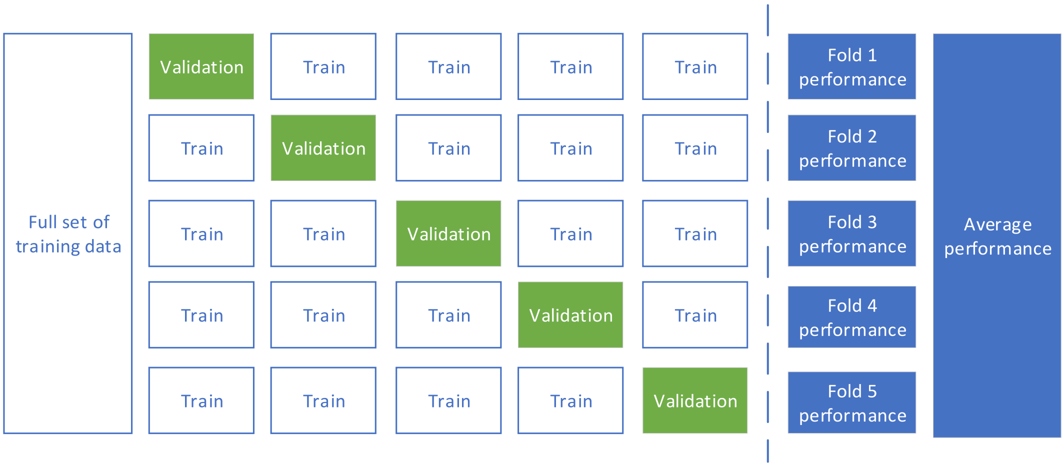


Fig. 15. Cross-validation using dataset folds in a ratio of 80:20.

validation set, and the other folds are used to train the model. This procedure is iterated several times, with a distinct fold serving as the validation set each time. To generate a more robust estimate of the model's performance, the outcomes from each validation step are finally averaged. The approach was performed with an 80:20 training-validation split, resulting in the creation of five distinct data folds, and therefore five distinct models were trained described in Fig. 15.

Notably, in this part of the experiment, the background was considered as first class. This consideration of the background class significantly contributes to the overall improvement in model performance, as background segmentation accuracy is usually particularly high. The results of the cross-validation are shown in Table 2. The Jaccard index coefficient metric was used to quantify model performance.

4.2.2. Statistical analysis of model architectures using ANOVA and Tukey HSD test

Since we could not compare the models trained on our image database for segmentation and classification of eye diseases with the reference results because they simply do not exist, we analyzed the results with statistical tests ANOVA [56] and the Tukey HSD [57] test to confirm their adequacy.

Prior to further statistical analyses we conducted a QQ-Plot (Quantile-Quantile Plot) for Standardized Residuals to assess the normality of the data distribution since the normal distribution of residuals is a common assumption in ANOVA [56] and the Tukey HSD [57] test as well as in many other statistical models.

The QQ-Plot visually compares the distribution of standardized residuals from our data to a theoretically normal distribution. Ideally, for the residuals to be considered normally distributed, the plot's points should align closely with the 45-degree reference line. In our analysis, the points from the QQ-Plot demonstrated a satisfactory alignment with this line, indicating that the assumptions of normality are reasonably met. The QQ-plot following our research is shown in Fig. 16.

Based on the results obtained from the QQ-Plot analysis, which suggested that our data distribution adheres to the normality assumption, we proceeded with a one-way Analysis of Variance (ANOVA) to evaluate the impact of the proposed network architecture modifications on the segmentation results. ANOVA is a statistical method used to compare the means of three or more samples to understand if at least one sample mean is significantly different from the others due to a specific factor. In this case, we compare the means of the five samples where factor is the architecture modification, and the samples are the Jaccard index coefficient from each fold. The null hypothesis asserts that all group means are equivalent, while the alternative hypothesis suggests that at least one group differs significantly.

To compute the p-value, we conducted the ANOVA test [58,59] by first calculating the mean Jaccard index coefficient for each of the five groups: U-Net, U-Net + ResNet34, U-Net + Inception V3, U-Net + VGG16, and U-Net + EfficientNet B3, based on their performance across all folds.

The next step, Between-Group Variance, involves calculating how much the group means deviates from the overall mean of all samples combined. This metric reflects the variance resulting from the differences in network architectures.

After Between-Group Variance, Within-Group Variance is calculated. Here, the variability of individual group samples around their respective group means is assessed. This variance measures the characteristic variation in segmentation performance within the same network architecture across different folds.

By dividing the between-group variance by the within-group variance the F-value is calculated. A higher F-value suggests a greater degree of difference among group means relative to the variation within groups. Finally, the p-value is derived from the F-distribution, based on the calculated F-value and the degrees of freedom associated with the between-group and within-group variances. The p-value indicates the probability of observing the calculated level of variation among group means if the null hypothesis were true.

A p-value less than 0.05 is a widely accepted standard indicating that the observed data would be quite unlikely under the null hypothesis, suggesting a real effect or difference between the groups tested. In our case, the calculated p-value of $5.294 \cdot 10^{-13}$ indicates that there is a statistically significant difference between the group means being compared in the ANOVA test. This result strongly suggests rejecting the null hypothesis that all group means are equal. Given this significant finding, we are going to make the Tukey Honest Significant Difference (HSD) test as the next step.

The Tukey HSD test is a widely endorsed method for this purpose, allowing for multiple pairwise comparisons while controlling for the family-wise error rate. This test is particularly useful in situations where multiple comparison tests are required, as it adjusts the p-values to account for the fact that the more tests you perform, the higher the chance of encountering a false positive. The outcome of the Tukey HSD test provides us with a means to rigorously compare each model against the others, returning pairwise significance levels and confidence intervals for the difference in means. These statistical tools guide us in determining which architectural enhancements to the U-Net model yield a statistically reliable improvement in segmentation performance. The analysis, again, was performed using the Jaccard Index as the primary metric for evaluating model performance (Table 2). The comparative study included

Table 2
Results of a 5-fold cross-validation with a Jaccard index coefficient in percentage as a reference with background class included.

Architecture	Fold 1 model IoU (%)	Fold 2 model IoU (%)	Fold 3 model IoU (%)	Fold 4 model IoU (%)	Fold 5 model IoU (%)
U-Net	53.0	54.2	50.1	52.0	52.2
U-Net + ResNet34	**73.2**	**72.6**	72.3	**71.2**	**76.0**
U-Net + Inception V3	69.5	68.6	69.7	67.5	69.7
U-Net + VGG16	65.3	69.5	65.2	64.3	65.5
U-Net + EfficientNet B3	67.2	69.3	**73.4**	70.6	68.0

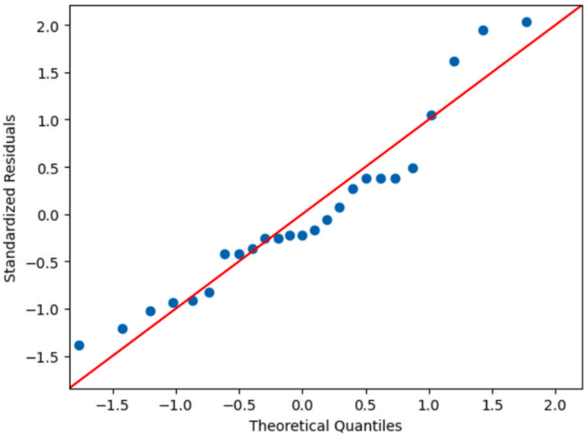


Fig. 16. Shows a QQ-plot, comparing the standardized residuals of a model to a theoretical normal distribution. The data aligns with the expected normal line, with minimal tailing observed, indicating a generally normal distribution of residuals.

combinations of the U-Net model with various architectures such as ResNet34, Inception V3, VGG16, and EfficientNet B3, each evaluated using a combination of Dice and Focal Loss functions. The results of Tukey HSD test are shown in Table 3.

In this comparative analysis, we specifically focused on how modifications to the U-Net architecture influence its segmentation accuracy, with a use of Jaccard Index. The p-values reported by the Tukey HSD test, as shown in Table 3, were derived through a comparison of the mean Jaccard Index scores across each unique pair of architectural configurations. The Tukey HSD test incorporates a statistical adjustment for multiple comparisons, effectively controlling the family-wise error rate. This adjustment is essential because the likelihood of encountering a type I error (false detecting a significant difference) increases with the number of pairwise comparisons conducted. The Tukey HSD method achieves this by utilizing the studentized range distribution to calculate a critical value that the observed mean differences must met for the differences to be deemed statistically significant. All comparisons between the base U-Net model and its enhancements with ResNet34, Inception V3, VGG16, and EfficientNet B3 demonstrate highly significant differences, with p-values of 0.001. This indicates that adding these architectures to U-Net notably improves its performance, underscoring the effectiveness of these modifications.

The comparisons within the enhanced U-Net models reveal varying degrees of performance impact. The match between U-Net + ResNet34 against U-Net + Inception V3 with resulting p-value of 0.013 and U-Net + VGG16 with p-value of 0.001 shows significant performance differences, highlighting a robust performance distinction. However, the comparison between U-Net + ResNet34 and U-Net + EfficientNet B3 yields a p-value of 0.046, situating it at the border of statistical significance. This suggests a meaningful but less distinct contrast in their performance, indicating that while different, the performance improvements from ResNet34 and EfficientNet B3 are closer than with others.

Worth mentioning, the comparison between U-Net + Inception V3 and U-Net + EfficientNet B3, with a p-value of 0.9, suggests no significant difference in their performance enhancements. This outcome implies that both architectural modifications offer comparable benefits. Additionally, the near-significant p-value of 0.09 for U-Net + Inception V3 versus U-Net + VGG16 indicates a slight but not statistically significant difference in performance, suggesting their potential equivalence in effectiveness.

The results of statistical analysis indicate that the U-Net + ResNet34 models besides consistent well performance, significantly differ from other models based on the different architecture, therefore further experiments will include U-Net + ResNet34 model alone.

4.2.3. The comparison of 5-fold cross-validation U-net + ResNet34 models

U-Net + ResNet34 models showed the best segmentation and classification performances on our image database. In Table 4 a mean

Table 3
Comparison of models using Jaccard Index based on different architectures with combination of Dice and Focal Loss using Tukey HSD test showing p-value for each pair combination.

Architecture 1	Architecture 2	p-value
U-Net	U-Net + ResNet34	0.001
U-Net	U-Net + Inception V3	0.001
U-Net	U-Net + VGG16	0.001
U-Net	U-Net + EfficientNet B3	0.001
U-Net + ResNet34	U-Net + Inception V3	0.013
U-Net + ResNet34	U-Net + VGG16	0.001
U-Net + ResNet34	U-Net + EfficientNet B3	**0.046**
U-Net + Inception V3	U-Net + VGG16	**0.09**
U-Net + Inception V3	U-Net + EfficientNet B3	**0.9**
U-Net + VGG16	U-Net + EfficientNet B3	0.028

Table 4
5-fold cross-validation results of U-Net + ResNet34 models based on the Jaccard and Dice Index. The best results are marked in bold, and average in bold italic.

Model	Jaccard Index (mean ± std)	DSC (mean ± std)
U-Net + ResNet34 Cross-Validation Model 1	0.7316 ± 0.1419	0.8375 ± 0.0916
U-Net + ResNet34 Cross-Validation Model 2	0.7259 ± 0.1721	0.8297 ± 0.1165
U-Net + ResNet34 Cross-Validation Model 3	0.7225 ± 0.1593	0.8288 ± 0.1096
U-Net + ResNet34 Cross-Validation Model 4	0.7122 ± 0.188	0.8172 ± 0.1351
U-Net + ResNet34 Cross-Validation Model 5	**0.7599 ± 0.1075**	**0.8596 ± 0.0656**
Average	***0.7398 ± 0.1575***	***0.834 ± 0.097***

score with standard deviations for both the Jaccard Index and DSC Index are presented for U-Net + ResNet34 models tested on each of 5 folds and trained on the remaining folds. The U-Net + ResNet34 Cross-validation Model 5 showed the best performance in respect to both metrics with relatively low standard deviations. All models have performance within the interval between 71 % and 75 % for Jaccard Index, and between 82 % and 86 % for DCS Index with relatively low standard deviations, which is a performance deviation of at most 4 % across different folds.

In addition, Precision, Recall, and Pixel Accuracy were evaluated, which provides a comprehensive understanding of model performance, as shown in Table 5. Best results for Precision, Recall, and Pixel Accuracy are indicated by U-Net + ResNet34 Cross-validation Models 2, 5, and 4 respectively.

In Table 6 sample confusion matrix, formed for U-Net + ResNet34 Cross-Validation Model 2, is presented and will be further examined. This matrix offers valuable insights into the model's ability to classify different classes and errors that the model makes in the classification by replacing one class with another. The highest values on the diagonal of the confusion matrix indicate the ability of the model to correctly identify instances of a particular class.

A confusion matrix provides a practical way to compare a model's performance across different classes, especially when there's an imbalance in the number of pixels belonging to each class. The background has very high precision and it has less than 1 % FP in favor of Corneal Edema. Epiphora class has the best performance with only 3 % belonging to the Epiphora class but being classified wrongly as belonging to the Background. Other classes have fewer pixels classified correctly and are only confused with the background. Zeroes implies that the model does not confuse samples originally belonging to disease class with each other, i.e. the classification boundary between disease classes was learned well by the model. The Cherry Eye prediction outcomes should be the focus for enhancing this model's performance. The classifier misclassified 30 % of samples which is the highest misclassification rate of all the classes. This pattern suggests difficulties in differentiating Cherry Eye from the background, potentially due to texture and color similarities. It's important to note that the confusion matrix in Table 6 represents a single model within our cross-validation framework, and outcomes may vary across models, sometimes favoring Cherry Eye detection over other classes. To enhance performance, we might consider modifying the model's architecture or training approach to distinguish Cherry Eye features more accurately. However, enhancing the detection of one condition could inadvertently affect the model's ability to recognize other conditions, indicating a trade-off in performance improvements. Our findings suggest that the model's primary challenge lies in distinguishing certain conditions from the background, rather than differentiating among conditions themselves.

4.2.4. *Example of qualitative analysis of U-net + ResNet34 model segmentation results*

In Fig. 17 examples of dog eye closeups and in Fig. 18 examples of whole head segmentations are shown. The columns Fig. 17(c1) and Fig. 18(c1) represent original images, columns Fig. 17(c2) and Fig. 18(c2) ground truth and columns Fig. 17(c3) and Fig. 18(c3) predictions using the U-Net + ResNet34 DL + FL model. Each row Fig. 17(r1)(r2)(r3) and Fig. 18(r1)(r2)(r3) describe an individual case. The classes are described using different colors where dark blue represents the background, light blue Corneal Edema, green Episcleral Congestion, orange Epiphora, and red represents Cherry Eye.

5. Discussion

From the resulting images of the test dataset, it is evident that the prediction model works within the scope of the expectation, providing the user with sufficient data to assist in the diagnosis of four eye conditions a pet can experience. The epiphora is the class

Table 5
Precision, Recall, and Pixel Accuracy evaluation of cross-validation models. The best results are marked in bold. Average results are written in bold italic.

Model	Precision (mean ± std)	Recall (mean ± std)	Pixel Accuracy (Mean)
U-Net + ResNet34 Cross-Validation Model 1	0.8271 ± 0.1332	0.868 ± 0.1007	0.9658
U-Net + ResNet34 Cross-Validation Model 2	**0.8715 ± 0.1045**	0.8137 ± 0.1718	0.981
U-Net + ResNet34 Cross-Validation Model 3	0.8166 ± 0.1683	0.8763 ± 0.1026	0.9585
U-Net + ResNet34 Cross-Validation Model 4	0.8317 ± 0.1083	0.8493 ± 0.2122	**0.9823**
U-Net + ResNet34 Cross-Validation Model 5	0.8235 ± 0.0997	**0.9125 ± 0.0862**	0.963
Average	***0.8331 ± 0.1134***	***0.863 ± 0.1165***	***0.9716***

Table 6
Confusion matrix of U-Net + ResNet34 Cross-Validation Model 2.

Class	Predicted background (%)	Predicted Corneal Edema (%)	Predicted Episcleral Congestion (%)	Predicted Epiphora (%)	Predicted Cherry Eye (%)
Background	**99**	1	0	0	0
Corneal Edema	15	**85**	0	0	0
Episcleral Congestion	16	0	**84**	0	0
Epiphora	3	0	0	**97**	0
Cherry Eye	30	0	0	0	**70**

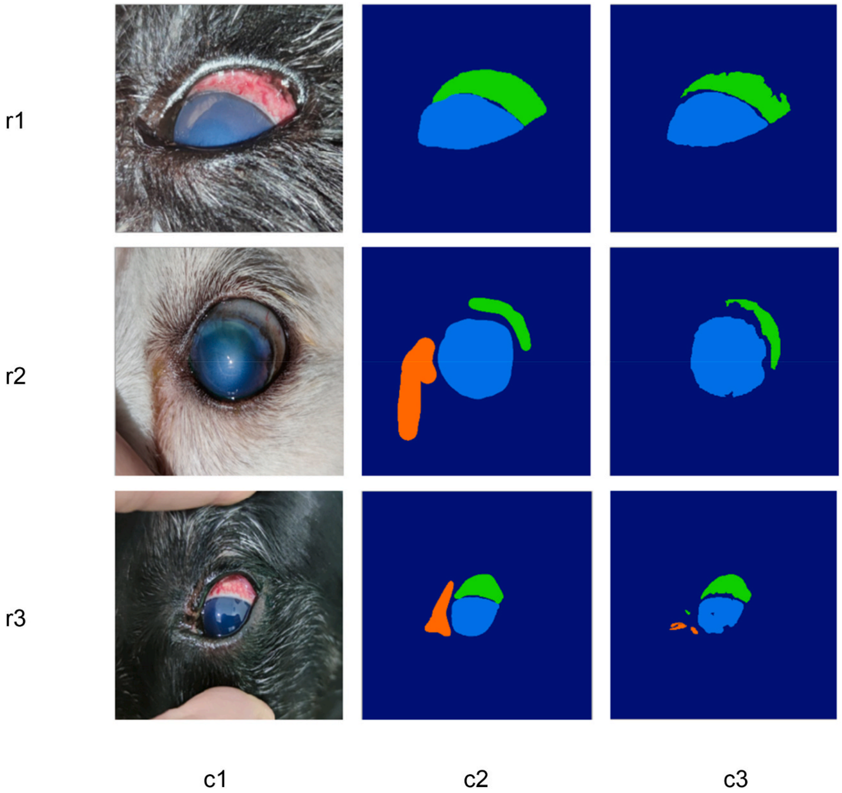


Fig. 17. Segmentation of test images containing closeup of an eye with diagnosis. The left column (c1) shows original images, the middle column (c2) shows ground truth (GT) and the right column (c3) predictions. The upper row (r1) predicts two classes as ground truth, the middle row (r2) fails to detect Epiphora class, and the lower row (r3) falsely detects another instance of Episcleral congestion.

which in general has proven to be the hardest to segment, even though U-Net + ResNet34 Cross-Validation Model 2, which was taken as an example shows otherwise, is also the most difficult to annotate due to not-so-distinguished boundaries. Considering the visual difference, when observed by the human eye, the classes that get most mixed between are the Cherry Eye and episcleral congestion. These are the ones developed models also find sometimes difficult to distinguish, especially when the observed object is small and distant. The problem that arises when the head figure or whole body is present in the image is related to false positives in regions that don't include the eye at all. Such a behavior can be prevented by focusing on an eye before starting a segmentation. The reason for the inclusion of these images in the dataset is the fact that dogs aren't always cooperative during the close-up eye image capturing. Further research could include handling mechanisms of obtaining closeup images by extracting regions with an eye and making an automatic selection of the best eye image in the image sequence. When comparing the results of models based on different backbones, datasets, and loss functions ResNet34 was demonstrated to be the optimal solution providing the best results in the majority of tests. Considering the time of test prediction execution, which is in close relation to complexity and resource requirements, in combination with other scores U-Net + ResNet34 was superior for the analysis of dataset from this study. Only base U-Net and U-Net + VGG16 have a faster response but at the considerable cost of a prediction score. Slightly better segmentation results of corneal edema have U-Net + Inception V3 which can be assumed is caused by better segmentation of images with a variable size of the cornea. The potential model improvement could be achieved by combining various backbones to keep practicality as the major priority.

The background should be discarded in average scores. The decision to exclude the background class from average scores warrants

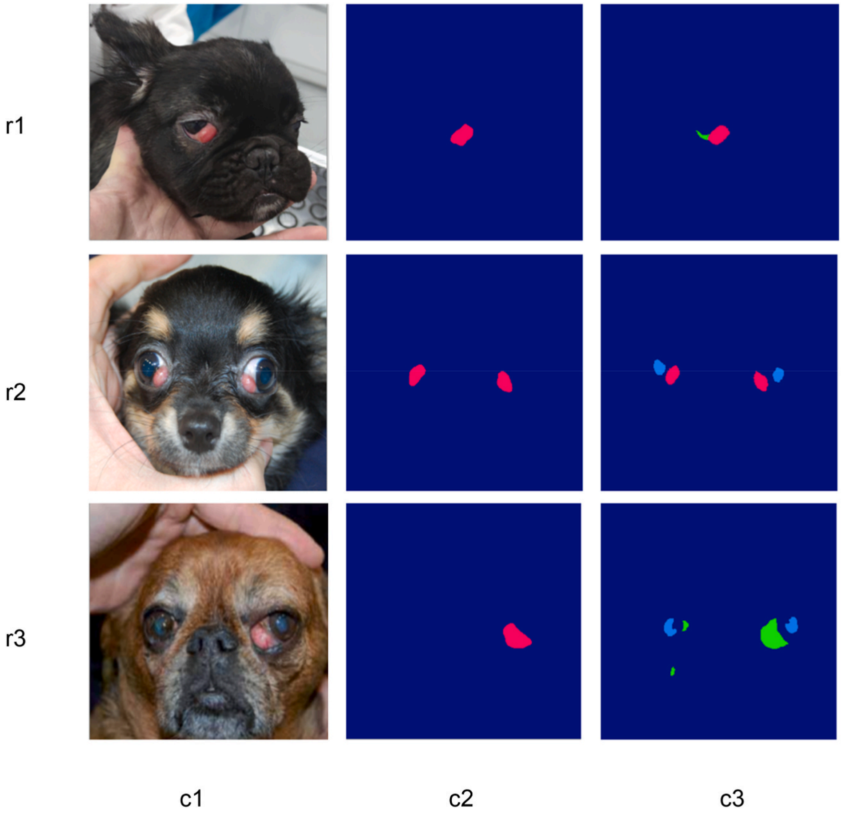


Fig. 18. Segmentation of test images containing whole head figure with diagnosis. The left column (c1) shows original images, the middle column (c2) ground truth (GT) and the right column (c3) predictions. The upper row (r1) predicts Cherry Eye positively and episcleral congestion falsely. The middle row (r2) makes a false detection of the corneal edema but correctly segments Cherry Eye. The lower row (r3) detects a combination of episcleral congestion and corneal edema classes where only the Cherry Eye class is present.

further elaboration. In this image segmentation study, it became evident that including the background class in our evaluation metrics did not significantly contribute to the practical insights gained from the model. The primary reason for discarding the background class in our average scores is twofold. Firstly, background pixels typically dominate the majority of images in the dataset, resulting in a disproportionately large share of the evaluation. However, these background regions often do not contain meaningful information related to the specific eye conditions under investigation. As a result, their inclusion would unduly influence the overall segmentation scores without providing valuable insights into the model's ability to differentiate between the classes of interest. Secondly, the study focuses on the accurate identification and segmentation of specific eye conditions in canines, a task that necessitates precise recognition of pathological features within the images. In most cases, the background class does not offer relevant information for the diagnostic process. Therefore, considering the high values achieved in the absence of the background class, its inclusion would not only deceive the observer but also complicate the interpretation of the model's performance on the classes of actual clinical interest. In essence, the exclusion of the background class from our average scores aligns with the study's primary objective, which is to assess the model's performance in identifying and segmenting Corneal Edema, Episcleral Congestion, Epiphora, and Cherry Eye. By doing so, the aim is to provide a more focused and accurate evaluation of the model's practicality and effectiveness in the domain of veterinary ophthalmology. By removing the background class from our average scores, we ensure that evaluation metrics provide a clear and meaningful reflection of the model's ability to address the specific challenges presented by canine ocular diseases. This deliberate choice emphasizes commitment to a balanced and informative assessment of the model's diagnostic capabilities.

The Jaccard and Dice Index are used as the main evaluation over other metrics such as Recall and Precision considering segmentation evaluation best practice [53].

The decision to use a combination of DL and FL gave better results than using FL, DL, or CCE alone. The choice to combine DL and FL, rather than employing each loss function individually, stemmed from the unique advantages they bring. Dice Loss, known for its ability to handle class imbalance effectively, excels in providing precise segmentation results. Focal Loss, on the other hand, prioritizes challenging examples during training, contributing to better convergence and performance in the presence of abundant easy examples. This combination effectively balanced the trade-off between high-precision segmentation and the ability to tackle complex cases. While the alternative strategy of combining FL and CCE was not specifically tested in this study, it can be reasonably assumed that the joined benefits of DL and FL would outperform using FL alone or in conjunction with CCE. This is due to the similarity in objectives between FL and CCE, with FL enhancing the concepts of CCE.

The inclusion of an additional 25 % of augmented images in the dataset proved to be a successful strategy in this research. By doing so, the dataset was extended to an optimal number of images based on [38]. It would be advisable to increase the number of input images to confirm this hypothesis in this case and will most probably be implemented in further research along with a higher number of classes.

During the training process, it is apparent that all models were converging as expected by continued lowering of the loss and an increase of IoU. It can be speculated that further training could result in better performance, but it would require significant resources, which might not be justified. It would take significantly more training to achieve just a mild improvement. Since the goal of this study was to make the model widely applicable, further research will be oriented toward making it more practical and easily applicable on mobile devices. The final model should strive to be a tradeoff between accuracy and accessibility.

Acknowledging the limitations and weaknesses of our study enriches its context and guides future research directions. The current dataset, though informative, could be enhanced by increasing its size and diversity with more classes, potentially improving model generalization. Incorporating attention mechanisms might also refine the results by focusing on relevant features. However, dataset expansion is challenging due to the extensive expert hours required for accurate annotation. Experimenting with a broader range of models and continuously enriching the dataset with more classes and images would likely enhance performance, despite the significant effort and time investment involved. Additionally, exploring advanced data analysis techniques could unveil deeper insights, further advancing the study's contributions.

Our model, originally designed for a canine eye disease, was applied to a human eye segmentation dataset – Open Eye Dataset (OpenEDS) [15]. This dataset describes three main components of an eye: the sclera, iris, pupil, and the background. Employing an architecture adjusted for canine datasets, the achieved mean Intersection-over-Union (mIoU) of 0.9503 closely approaches the leading score of 0.95276 by RIT-MVRL [60], significantly exceeding the established baseline of 0.89 [15]. This performance not only underscores the model's robustness but also its potential adaptability for a broader array of segmentation tasks beyond its initial veterinary diagnostic focus.

Notably, the absence of parallel research efforts specifically addressing canine eye segmentation highlights the innovative application of this model. The minor practical difference in mIoU relative to the challenge's top entry suggests the possible further model refinement, especially with a specific segmentation task in mind. Despite the model's origination for diagnosing canine eye diseases, its competitive performance in a human eye segmentation challenge illustrates its broader applicability and underscores the potential for cross-species diagnostic tools. While there is a limited supply of datasets and study dedicated to segmenting the human eyes for numerous reasons, corresponding datasets and studies focusing on canine subjects are virtually nonexistent. The scarcity of comparable studies, especially in the segmentation of canine ocular diseases, highlights the unique contribution of this research and its potential impact on the field.

Based on the research results, we suggest further exploration and development of the U-Net + ResNet34 model. This proposed model is designed for the recognition and pixel-based localization of clinical symptoms and canine eye diseases, with a primary focus on achieving the highest Intersection-over-union (IoU) scores for individual and combined classes. The adjustments made to enhance robustness and practicality aim to facilitate effective decision-making for non-specialized veterinary applications.

6. Conclusion

In the domain of veterinary ophthalmology, this research embodies both promise and challenge. The author's machine learning model for diagnosing ocular diseases in canines indicates precision and accessibility, redefining the boundaries of diagnostic capability.

The model's diagnostic accuracy is essential to this achievement, reflected in Intersection over Union (IoU) scores that surpass 80 % for a particular class. It empowers veterinarians and pet owners with a dependable, user-friendly tool for the early detection of ocular diseases. This decentralization of diagnosis paves the way for improved treatment outcomes, which will eventually promote the well-being of canine patients.

Acknowledging these leaps, it is vital to confront the inherent constraints that frame this study. The dataset, while carefully assembled, remains but a fragment of the expansive clinical landscape. Therefore, as in computer vision tasks, the pursuit of larger and more diverse datasets is imperative to augment the model's generalizability.

Furthermore, the specter of subjectivity in data interpretation, despite expert oversight, may benefit from collaboration with a panel of specialists to mitigate individual bias. The exploration of alternative hyperparameter configurations remains pivotal for enhancing the model's precision and robustness.

A notable constraint comes in the form of scalability. The dataset needs updating and rebalancing with each new disease that justifies inclusion. This task proves complex, as diseases occur with varying frequencies, potentially making dataset maintenance an ongoing challenge.

Real-world application and seamless integration into clinical settings represent uncharted terrain. Bridging the gap between model development and practical deployment remains a focal point. Simultaneously, the constant need for vigilance is required by ethical and privacy concerns, ensuring compliance with legal and moral standards.

The path ahead offers numerous avenues for refinement.

- Collaboration with Specialists: Engaging a panel of veterinary experts further elevates data interpretation and model calibration.
- Hyperparameter Exploration: Exhaustive hyperparameter exploration can result in more precise and adaptable models.
- Real-World Integration: Integration into clinical systems and real-world deployment demands exploration.

- Ethical and Regulatory Adherence: Ongoing commitment to ethical and regulatory compliance is imperative.

In sum, this research embodies a juncture where technology converges with veterinary practice, elevating the skill of ocular disease diagnosis. In this context, benefits, limitations, and the path ahead converge to extend the reach and influence of veterinary ophthalmology.

Data availability statement

The Dog Eye Segmentation 4-Class Ophthalmic Disease (DogEyeSeg4) Dataset: URN:NBN: HR:195:405214 associated with the submitted journal article is publicly available for further research at the [41]. The dataset is under copyright protection and is available for use according to the "In Copyright" terms specified by the International Rights Statement. Researchers are encouraged to adhere to these copyright restrictions when using the dataset.

Ethics and consent statement

The images included in the dataset were collected as part of routine clinical evaluations conducted by a veterinary ophthalmology specialist. For the purposes of this study, images were selected randomly and anonymized, ensuring that no examination dates, client information, or animal identifiers were included. Given that the images were obtained during standard medical care and no animals were harmed for the purpose of image acquisition, approval by an ethics committee was not required for this study.

CRediT authorship contribution statement

Matija Buric: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sinisa Grozdanic:** Validation, Funding acquisition, Data curation. **Marina Ivasic-Kos:** Writing – review & editing, Validation, Supervision, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:Marina Ivasic-Kos reports financial support was provided by Animal Eye Consultants of Iowa, Hiawatha, IA, USA. Sinisa Grozdanic reports a relationship with Animal Eye Consultants of Iowa, Hiawatha, IA, USA that includes: board membership and employment.

Acknowledgments

This research was supported by an Image-based AI-assisted diagnostics of canine ocular disease (AICODD) project funded by Animal Eye Consultants of Iowa. Iowa, USA.

References

[1] P. Wang, N. Vasconcelos, Towards professional level crowd annotation of expert domain data, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Vancouver, BC, Canada, Jun. 2023, pp. 3166–3175, https://doi.org/10.1109/CVPR52729.2023.00309.

[2] N. Seliya, A. Abdollah Zadeh, T.M. Khoshgoftaar, A literature review on one-class classification and its potential applications in big data, J Big Data 8 (1) (Sep. 2021) 122, https://doi.org/10.1186/s40537-021-00514-x.

[3] Y. Huang, et al., A multi-label learning prediction model for heart failure in patients with atrial fibrillation based on expert knowledge of disease duration, Appl. Intell. 53 (17) (Sep. 2023) 20047–20058, https://doi.org/10.1007/s10489-023-04487-7.

[4] S. Grozdanić, S. Đukić, S. Luzhetskiy, N. Milčić-Matić, T. Lazić, Atlas bolesti oka pasa i mačaka, Beograd: Oculus Vet (2020).

[5] F. Abdullah, et al., A review on glaucoma disease detection using computerized techniques, IEEE Access 9 (2021) 37311–37333, https://doi.org/10.1109/ACCESS.2021.3061451.

[6] Y. Hagiwara, et al., Computer-aided diagnosis of glaucoma using fundus images: a review, Comput. Methods Progr. Biomed. 165 (Oct. 2018) 1–12, https://doi.org/10.1016/j.cmpb.2018.07.012.

[7] M.S. Haleem, L. Han, J. van Hemert, B. Li, Automatic extraction of retinal features from colour retinal images for glaucoma diagnosis: a review, Comput. Med. Imag. Graph. 37 (7–8) (Oct. 2013) 581–596, https://doi.org/10.1016/j.compmedimag.2013.09.005.

[8] L. Li, M. Xu, X. Wang, L. Jiang, H. Liu, Attention based glaucoma detection: a large-scale database and CNN model, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, Jun. 2019, pp. 10563–10572, https://doi.org/10.1109/CVPR.2019.01082.

[9] N. Chakrabarty, S. Chatterjee, A novel approach to glaucoma screening using computer vision, in: *2019 International Conference On Smart Systems And Inventive Technology (ICSSIT)*, Tirunelveli, India, IEEE, Nov. 2019, pp. 881–884, https://doi.org/10.1109/ICSSIT46314.2019.8987803.

[10] Zhuo Zhang, et al., ORIGA^{-light}: an online retinal fundus image database for glaucoma analysis and research, in: 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology, IEEE, Buenos Aires, Aug. 2010, pp. 3065–3068, https://doi.org/10.1109/IEMBS.2010.5626137.

[11] S. Kanse, D. Yadav, Retinal fundus image for glaucoma detection: a review and study, J. Intell. Syst. 28 (Jan. 2019) 43–56, https://doi.org/10.1515/jisys-2016-0258.

[12] J. Sivaswamy, S.R. Krishnadas, G. Datt Joshi, M. Jain, A.U. Syed Tabish, Drishti-GS: retinal image dataset for optic nerve head(ONH) segmentation, in: 2014 IEEE 11th International Symposium on Biomedical Imaging (ISBI), Apr. 2014, pp. 53–56, https://doi.org/10.1109/ISBI.2014.6867807.

[13] F. Fumero, S. Alayón, J.L. Sanchez, J. Sigut, M. Gonzalez-Hernandez, RIM-ONE: an open retinal image database for optic nerve evaluation, presented at the Int. Sym. on CBMS (Jul. 2011) 1–6, https://doi.org/10.1109/CBMS.2011.5999143.

[14] B. Luo, J. Shen, Y. Wang, M. Pantic, The iBUG Eye Segmentation Dataset, 2019, p. 9, https://doi.org/10.4230/OASICS.ICCSW.2018.7.

[15] S.J. Garbin, Y. Shen, I. Schuetz, R. Cavin, G. Hughes, S.S. Talathi, OpenEDS: open eye dataset, arXiv: arXiv:1905.03702 (2019) [Online]. Available: http://arxiv.org/abs/1905.03702. (Accessed 22 November 2022).

[16] R. Bernardes, P. Serranho, C. Lobo, Digital ocular fundus imaging: a review, Ophthalmologica 226 (4) (2011) 161–181, https://doi.org/10.1159/000329597.

[17] H. Proenca, S. Filipe, R. Santos, J. Oliveira, L.A. Alexandre, The UBIRIS.v2: a database of visible wavelength Iris images captured on-the-move and at-a-distance, IEEE Trans. Pattern Anal. Mach. Intell. 32 (8) (Aug. 2010) 1529–1535, https://doi.org/10.1109/TPAMI.2009.66.

[18] H. Hofbauer, F. Alonso-Fernandez, P. Wild, J. Bigun, A. Uhl, A ground truth for Iris segmentation, in: 2014 22nd International Conference on Pattern Recognition, IEEE, Stockholm, Sweden, Aug. 2014, pp. 527–532, https://doi.org/10.1109/ICPR.2014.101.

[19] W. Fuhl, G. Kasneci, E. Kasneci, TEyeD: over 20 million real-world eye images with pupil, eyelid, and Iris 2D and 3D segmentations, 2D and 3D landmarks, 3D eyeball, gaze vector, and eye movement types, in: 2021 IEEE International Symposium on Mixed and Augmented Reality (ISMAR), Oct. 2021, pp. 367–375, https://doi.org/10.1109/ismar52148.2021.00053.

[20] M. Buric, B. Kovacic, M. Ivasic-Kos, The dog eye guardian app: from image to diagnosis with AI insights, in: *2024 9th International Conference On Smart And Sustainable Technologies (SpliTech)*, Bol and Split, Croatia, IEEE, Jun. 2024, pp. 1–6, https://doi.org/10.23919/SpliTech61897.2024.10612583.

[21] B.J. Antony, et al., A combined machine-learning and graph-based framework for the segmentation of retinal surfaces in SD-OCT volumes, Biomed. Opt Express 4 (12) (Dec. 2013) 2712, https://doi.org/10.1364/BOE.4.002712.

[22] J. Liu, A. Kanazawa, D. Jacobs, P. Belhumeur, "Dog breed classification using Part Localization," in computer vision – eccv 2012, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2012, pp. 172–185, https://doi.org/10.1007/978-3-642-33718-5_13.

[23] X. Wang, V. Ly, S. Sorensen, C. Kambhamettu, Dog breed classification via landmarks, in: 2014 IEEE International Conference on Image Processing, ICIP, Oct. 2014, pp. 5237–5241, https://doi.org/10.1109/ICIP.2014.7026060.

[24] S. Somppi, H. Törnqvist, L. Hänninen, C. Krause, O. Vainio, Dogs do look at images: eye tracking in canine cognition research, Anim Cogn 15 (2) (Mar. 2012) 163–174, https://doi.org/10.1007/s10071-011-0442-1.

[25] M. Buric, M. Pobar, M. Ivasic-Kos, Object detection in sports videos, in: *2018 41st International Convention On Information And Communication Technology, Electronics And Microelectronics (MIPRO)*, Opatija, IEEE, May 2018, pp. 1034–1039, https://doi.org/10.23919/MIPRO.2018.8400189.

[26] M.L. Mack, I. Gauthier, J. Sadr, T.J. Palmeri, Object detection and basic-level categorization: sometimes you know it is there before you know what it is, Psychonomic Bulletin & Review 15 (1) (Feb. 2008) 28–35, https://doi.org/10.3758/PBR.15.1.28.

[27] E. Cengil, A. Çinar, M. Yildirim, A case study: cat-dog face detector based on YOLOv5, in: 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Sep. 2021, pp. 149–153, https://doi.org/10.1109/3ICT53449.2021.9581987.

[28] O.M. Parkhi, A. Vedaldi, C.V. Jawahar, A. Zisserman, The truth about cats and dogs, in: 2011 International Conference on Computer Vision, Nov. 2011, pp. 1427–1434, https://doi.org/10.1109/ICCV.2011.6126398.

[29] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, arXiv: arXiv:1512.03385 (Dec. 10, 2015) [Online]. Available: http://arxiv.org/abs/1512.03385. (Accessed 3 November 2022).

[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv: arXiv:1409.1556 (2015) [Online]. Available: http://arxiv.org/abs/1409.1556. (Accessed 3 November 2022).

[31] J.Y. Kim, H.E. Lee, Y.H. Choi, S.J. Lee, J.S. Jeon, CNN-based diagnosis models for canine ulcerative keratitis, Sci. Rep. 9 (1) (2019), https://doi.org/10.1038/s41598-019-50437-0.

[32] C. Szegedy, et al., Going Deeper with Convolutions, vol. 16, 2014 arXiv: arXiv:1409.4842, http://arxiv.org/abs/1409.4842. (Accessed 4 November 2022).

[33] Y. LeCun, et al., Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (Dec. 1989) 541–551, https://doi.org/10.1162/neco.1989.1.4.541.

[34] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, arXiv: arXiv:1505.04597 (2015) [Online]. Available: http://arxiv.org/abs/1505.04597. (Accessed 3 November 2022).

[35] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (Jun. 2010) 303–338, https://doi.org/10.1007/s11263-009-0275-4.

[36] T.-Y. Lin, et al., "Microsoft COCO: common objects in context," in computer vision – ECCV 2014, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Lecture Notes in Computer Science, vol. 8693, Springer International Publishing, Cham, 2014, pp. 740–755, https://doi.org/10.1007/978-3-319-10602-1_48, 8693.

[37] A. Kuznetsova, et al., The Open Images Dataset V4: unified image classification, object detection, and visual relationship detection at scale, Int. J. Comput. Vis. 128 (7) (Jul. 2020) 1956–1981, https://doi.org/10.1007/s11263-020-01316-z.

[38] J. Deng, W. Dong, R. Socher, L.-J. Li, Kai Li, Li Fei-Fei, ImageNet: a large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Miami, FL, Jun. 2009, pp. 248–255, https://doi.org/10.1109/CVPR.2009.5206848.

[39] Y. Xiao, et al., A review of object detection based on deep learning, Multimed Tools Appl 79 (33–34) (Sep. 2020) 23729–23791, https://doi.org/10.1007/s11042-020-08976-6.

[40] O. Russakovsky, et al., ImageNet large scale visual recognition challenge, Int. J. Comput. Vis. 115 (3) (Dec. 2015) 211–252, https://doi.org/10.1007/s11263-015-0816-y.

[41] M. Burić, M. Ivašić-Kos, and S. Grozdanić, "DogEyeSeg4: Dog Eye Segmentation 4-Class Ophthalmic Disease Dataset." Faculty of Informatics and Digital Technologies, University of Rijeka Accessed: August. 22, 2024. [Online]. Available: https://urn.nsk.hr/urn:nbn:hr:195:405214.

[42] T. Nemoto, et al., Effects of sample size and data augmentation on U-Net-based automatic segmentation of various organs, Radiol. Phys. Technol. 14 (3) (Sep. 2021) 318–327, https://doi.org/10.1007/s12194-021-00630-6.

[43] L.F. Sánchez-Peralta, A. Picón, F.M. Sánchez-Margallo, J.B. Pagador, Unravelling the effect of data augmentation transformations in polyp segmentation, Int J CARS 15 (12) (Dec. 2020) 1975–1988, https://doi.org/10.1007/s11548-020-02262-4.

[44] W. Rawat, Z. Wang, Deep convolutional neural networks for image classification: a comprehensive review, Neural Comput. 29 (9) (Sep. 2017) 2352–2449, https://doi.org/10.1162/neco_a_00990.

[45] M. Tan, Q.V. Le, EfficientNet: rethinking model scaling for convolutional neural networks, arXiv: arXiv:1905.11946 (2020) [Online]. Available: http://arxiv.org/abs/1905.11946. (Accessed 3 November 2022).

[46] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2012. Nov. 10, 2022. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

[47] E. Nichani, A. Radhakrishnan, C. Uhler, Increasing depth leads to U-shaped test risk in over-parameterized convolutional networks, arXiv: arXiv:2010.09610 (2020) [Online]. Available: http://arxiv.org/abs/2010.09610. (Accessed 14 November 2022).

[48] R. Wang, et al., Deep residual network framework for structural health monitoring, Struct. Health Monit. 20 (4) (Jul. 2021) 1443–1461, https://doi.org/10.1177/1475921720918378.

[49] B. Singh, L.S. Davis, An analysis of scale invariance in object detection - SNIP, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, Jun. 2018, pp. 3578–3587, https://doi.org/10.1109/CVPR.2018.00377.

[50] N. Siddique, P. Sidike, C. Elkin, and V. Devabhaktuni, "U-net and its Variants for Medical Image Segmentation: Theory and Applications," p. 42.

[51] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines".

[52] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (Oct. 1986) 533–536, https://doi.org/10.1038/323533a0.

[53] J.S. Bridle, Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition, in: F.F. Soulié, J. Hérault (Eds.), Neurocomputing, Springer Berlin Heidelberg, Berlin, Heidelberg, 1990, pp. 227–236, https://doi.org/10.1007/978-3-642-76153-9_28.

[54] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, arXiv: arXiv:1505.00853 (2015) [Online]. Available: http://arxiv.org/abs/1505.00853. (Accessed 15 November 2022).

[55] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollar, Focal loss for dense object detection, in: 2017 IEEE International Conference on Computer Vision (ICCV), IEEE, Venice, Oct. 2017, pp. 2999–3007, https://doi.org/10.1109/ICCV.2017.324.

[56] A.F. Siegel, Anova, in: Practical Business Statistics, Elsevier, 2016, pp. 469–492, https://doi.org/10.1016/B978-0-12-804250-2.00015-8.

[57] J.W. Tukey, Comparing individual means in the analysis of variance, Biometrics 5 (2) (Jun. 1949) 99, https://doi.org/10.2307/3001913.

[58] A.F.M. Alkarkhi, W.A.A. Alqaraghuli, Easy Statistics for Food Science with R: Abbas F.M. Alkarkhi (Malaysian Institut of Chemical & Bioengineering Technology Universiti Kuala Lumpur, UniKL, MICET, 78000 Melaka, Malaysia), Wasin A.A. Alqaraghuli (Skill Education Center, PA, A-07-03 Pearl Avenue, Sungai Chua, 43000 Kajang, Selangor, Malaysia), Academic Press, London San Diego Cambridge Kidlington, 2019.

[59] D.C. Montgomery, Design and Analysis of Experiments, ninth ed., John Wiley & Sons, Inc, Hoboken, NJ, 2017.

[60] A.K. Chaudhary, et al., RITnet: real-time semantic segmentation of the eye for gaze tracking, in: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Oct. 2019, pp. 3698–3702, https://doi.org/10.1109/ICCVW.2019.00568.