Norwegian University
of Life Sciences

Master's Thesis 2024 30 ECTS
Faculty of Science and Technology (REALTEK)

Diagnostics of Canine Elbow Dysplasia using Deep Learning with Explainability Analysis

Artush Mkrtchyan
Data Science

Acknowledgements

This master's thesis represents not just the culmination of my academic journey in Data Science at NMBU over the past five years, but also an unexpected journey into the realm of veterinary science. It's a testament to life's unpredictability and its capacity to steer us towards paths unforeseen.

I am profoundly grateful to PhD candidate Bao Ngoc Huynh, whose expertise in machine learning and medical imaging has been invaluable. Her insightful pointers and hands-on approach have greatly enhanced my understanding and methodology throughout this semester.

Special thanks are due to Professor Cecilia Marie Futsæther, my main supervisor, for her unwavering guidance and support. Her ability to inspire and push me beyond my perceived limits has been pivotal in reaching the heights of this academic endeavor.

I extend my gratitude to Associate professor Hege Kippenes Skogmo, an international specialist in veterinary radiology at NMBU Faculty of Veterinary Medicine. Her discussions on the implications of our results have enriched the practical value of my thesis.

Additionally I must thank Associate Professor Oliver Tomic for his active engagement and insightful contributions during our project meetings.

Lastly, I am forever indebted to my fellow students at the study space, whose camaraderie made this rigorous journey a memorable and enjoyable experience. My deepest appreciation also goes to my parents, whose endless support has been my anchor throughout my life.

Thank you all for your part in my journey.

Abstract

Elbow Dysplasia is a term used to describe the presence of one or several abnormalities involving the elbow joint. It is genetically inheritable and could in the worst case lead to lameness in dogs. In Norway, dogs can be screened of elbow dysplasia when they are a minimum of 12 months old. This thesis builds on the NMBU master thesis by Steiro, where Steiro used convolutional neural networks on dog elbow x-ray images to automatically diagnose elbow dysplasia. This thesis delved therefore deeper, to see if one could identify the severity grade of elbow dysplasia by experimenting with parameters in convolutional neural networks.

There was a total of 7229 x-ray images collected from various clinics across Norway between 2018 and 2021, that were used for analysis in this thesis. EfficientNet models of different complexities and other parameters, such as type of loss function and learning rate were used for classification. There were four-class models which looked at one normal class and three classes of abnormal elbows with increasing severity, and three-class models which only looked at abnormal elbows, that were tested.

The highest overall performing model in terms of the four-class models, had a test accuracy of 95.8% and a high test MCC of 0.805. On the other hand, the highest overall performing three-class model had a test accuracy of 76.4% combined with a test MCC of 0.643.

In addition to experimenting with different loss functions and learning rates, two distinct pre-processing methods were tested to boost performance for the three-class models. The first technique was using images from three channels, where two of the channels had augmented versions of the original image. The second technique was binarization of the image dataset, where two of the three classes of abnormal elbows were merged, making it a binary problem.

Lastly, explainability analysis was implemented on the highest overall performing three-class model, to assess if the model could have potential to be used in a clinical setting. This was done with a method called Variance of the Gradients, to understand which regions of the elbow joint most affected the model's predictions. This method proved that the model was not reliable, because the model often looked outside of the elbow joint, which is outside the intended region of interest.

Contents

Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under ROC
B1 - B4	EfficientNet model complexities
ED	Elbow Dysplasia
FN	False Negatives
FP	False Positives
LR	Learning Rate
MCC	Matthews Correlation Coefficient
MCD	Fragmented Medial Coronoid Process
ML	Machine Learning
MSE	Mean Squared Error
NKK	Norsk Kennel Klub
OCD	Osteochondritis Dissecans
ROC	Receiving Operating Characteristic
ROI	Region of Interest
TN	True Negatives
TP	True Positives
UAP	Ununited Anconeal Process
VarGrad	Variance of Gradients

Chapter 1

Introduction

Elbow Dysplasia (ED) is a collective term encompassing various developmental abnormalities in the elbow joint of dogs, with genetic factors playing a significant role in its manifestation [1]. This condition can severely impair a dog's mobility, in the worst case leading to lameness if not addressed. ED is manually diagnosed with the use of x-ray images by veterinary radiologists. The diagnostic process can therefore be time consuming [1]. This thesis has its basis on work done by previous master's student Steiro [2], and explores the use of convolutional neural networks with a focus on classifying the grade of elbow dysplasia.

1.1 Motivation and related works

Elbow Dysplasia represents a significant challenge, particularly within canine breeding programs. This is relevant, especially in Norway where Norsk Kennel Klub (NKK), enforces strict regulations to mitigate the propagation of this condition [3]. The current diagnostic process involves manual evaluation by veterinary radiologists, who analyze x-ray images to classify the severity of ED on a scale from 0 to 3 [4]. This manual classification is time-intensive, taking approximately five minutes per evaluation, and must be performed by certified specialists, which there are only two of in Norway [4]. With about 4500 - 5000 dogs assessed annually, the process demands a substantial amount of specialist time and resources [5].

The development of automated tools, such as multi-class models for classifying the severity of ED, holds clinical potential. Take for instance the results from Steiro's thesis, where the classification time per x-ray image was about one second at most [2]. If such tools can operate both reliably and efficiently, they could reduce the time required for each diagnosis and potentially increase the accuracy of classifications. Such improvements could expedite the decision-making process in breeding programs and improve the quality of life for veterinarians, ultimately leading to healthier canine populations [6].

In terms of advancements in related fields, the study by Zhou et al. demonstrates the effectiveness of multi-task learning frameworks in human medicine, specifically for tumor classification and segmentation in 3D automated breast ultrasound images

[7]. This approach illustrates the potential for similar methodologies to be adapted for veterinary applications, suggesting that such tools could enhance the diagnostic process for conditions like ED.

Beyond this, the existing literature within veterinary medicine shows a growing interest in the application of machine learning techniques. For example, Boufenar et al. developed a deep learning model tailored for diagnosing canine hip dysplasia [8]. They achieved an accuracy score of 98.32%, with recall at 98.35% and precision of 98.44% [8]. Similarly, several examples of machine learning research in veterinary diagnostic imaging are summarized in a literature review by Hennessey et al. [9]. Examples include the use of deep learning models to analyze canine radiographs for identifying the maturity and timing of bone fractures, and a study comparing AI against human evaluations for measuring the vertebral heart scale in cats and dogs [9].

In human medicine, the efficacy of AI tools has been explored with promising results. Meetschen et al. (2024) highlighted how AI could assist radiology residents by improving fracture detection sensitivity and reducing interpretation times [10]. These examples from both human and veterinary medicine underscore the potential of AI to enhance diagnostic imaging across the medical field.

In summary, the motivation for this thesis stems from the need to improve and streamline the diagnostic process for elbow dysplasia in dogs. This work aims to explore and extend these emerging technologies, supporting breeding regulations and enhancing animal welfare by integrating advanced computational techniques into the diagnostic workflow.

1.2 Purpose

This master's thesis aims to enhance the classification performance of multi-class models in diagnosing canine elbow dysplasia, building on previous work by Steiro [2]. The goals are to improve the classification performance of the multi-class models, and to apply explainability analysis with the method Variance of the Gradients (VarGrad) [11], to understand which regions of the elbow joint most affected the model's predictions and see if the model has potential in a clinical setting.

EfficientNet, a type of pre-trained convolutional neural network [12], was used with various complexities and other parameters to classify x-ray images of canine elbows. This thesis includes four-class EfficientNet models that classify images into one of four classes (one class for normal elbows, and three classes of abnormal elbows with increasing severity), and three-class EfficientNet models that classify into one of the three classes of abnormal elbows. The highest overall performing three-class model had VarGrad employed to assess its predictions for explainability analysis.

To potentially enhance model performance, two distinct pre-processing methods were tested on the three-class models. These methods include binarization of the image dataset, merging two of the three classes of abnormal elbows, and using three image channels, incorporating augmented image data where two of the channels contain augmented versions of the original x-ray image.

There were two datasets utilized in this thesis, one for the four-class models and another for the three-class models. The four-class dataset consisted of 7229 x-ray images, of which 3030 were of abnormal elbows. These images were received by the veterinarians at NMBU Faculty of Veterinary Medicine in September 2022 and fully pre-processed in September 2023. The three-class dataset included the images of abnormal elbows only, and was made from a copy of the full dataset, but with all the cases of normal elbows removed, pre-processed in February 2024.

1.3 Layout

This thesis has the following layout: Chapter 2 outlines the theoretical framework underpinning the thesis. Chapter 3 describes the datasets and methodologies employed. Chapter 4 presents the analytical results, including metrics and plots. Discussion of these results and avenues for future research is covered in Chapter 5. Chapter 6 concludes the thesis while the final chapter lists all references cited and is followed by two appendices.

Chapter 2

Theory

2.1 X-ray imaging

X-ray imaging stands as a foundational diagnostic tool in medicine, employing x-rays, a type of electromagnetic radiation with wavelengths shorter than visible light, to explore the body's internal structure [13]. Discovered by Wilhelm Conrad Röntgen in 1895, this technology quickly became indispensable in medicine [14]. This method relies on attenuation, governed by Beer-Lambert's law, which describes how the x-ray beam's intensity diminishes as it traverses tissues of varying densities. Dense tissues like bones absorb more x-rays, thus appearing white on images, whereas softer tissues absorb less, resulting in darker shades of gray. This differential absorption provides clear contrast between various tissues, enabling the effective diagnosis of numerous conditions [13].

The utility of x-ray imaging spans identifying bone fractures, dental issues, and respiratory diseases like pneumonia [14]. It's pivotal in mammography for breast cancer screening, underscoring its role in preventive healthcare. Despite newer imaging technologies, x-ray's simplicity, speed, and affordability keep it indispensable in global healthcare practices. An example of an x-ray image is given in figure 2.1.

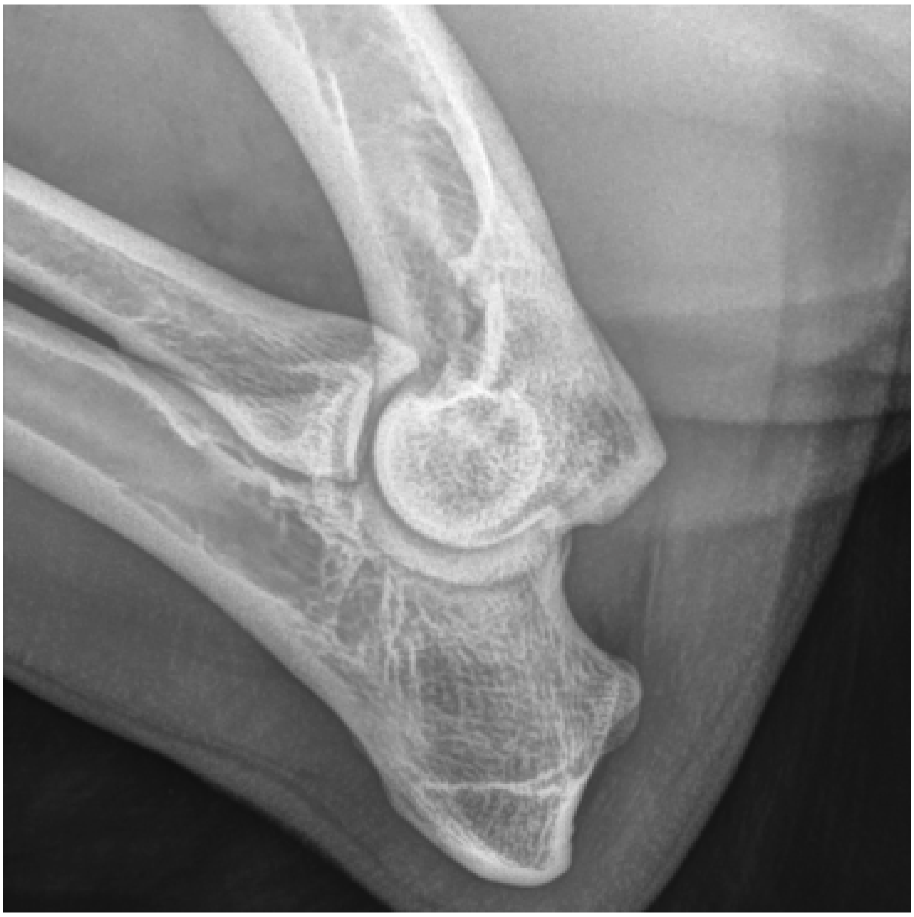


Figure 2.1: An x-ray image depicting a normal dog elbow.

2.2 Elbow dysplasia and diagnostics

Table 2.1: Elbow Dysplasia scoring: The Nordic countries adhere to criteria proposed by IEWG (International Elbow Working Group). Table based on Tellheim 2011 [15]

Elbow Dysplasia Scoring	**Radiographic Finding**
Normal elbow joint (grade 0)	Normal elbow joint with no evidence of incongruity, sclerosis or arthrosis
Mild arthrosis (grade 1)	Presence of osteophytes less than 2mm. Sclerosis of the base of the coronoid processes - trabecular pattern still visible
Moderate arthrosis or suspect primary lesion (grade 2)	Presence of osteophytes between 2 and 5 mm. Obvious sclerosis (no trabecular pattern) of the base of the coronoid processes. Step of 3 - 5 mm between radius and ulna (incongruity). Indirect signs for other primary lesion (UAP, MCD, OCD)
Severe arthrosis or evident primary lesion (grade 3)	Presence of osteophytes $>$ 5mm. Step of $>$ 5mm between radius and ulna (obvious incongruity). Obvious presence of primary lesion (UAP, MCD, OCD)

Elbow dysplasia in dogs involves developmental abnormalities of the elbow joint, encompassing the ulna, humerus, and radius [1]. These abnormalities, including osteochondritis dissecans (OCD), ununited anconeal process (UAP), fragmented medial coronoid process (FMCP or MCD), osteoarthritis (arthrosis), and increased bone density (sclerosis), disrupt joint function, leading to lameness [16]. Diagnosis, based on clinical examination and imaging like x-rays or CT scans, categorizes ED from normal to grade 3 (severe), as detailed in table 2.1. Furthermore, veterinary radiologists diagnose based on the most severe diagnosis [4]. Therefore, if a patient has arthrosis equivalent to an ED grade 1 diagnosis, but also obvious presence of UAP, the diagnosis automatically gets classified as ED grade 3 [4].

Veterinarians often use images of both elbows of the patients to diagnose ED [4]. This is since it can be hard to detect ED through x-ray images, because some of the abnormalities can look similar to the natural bone structure around the elbows [1]. Most often, if one elbow has a lesion, both elbows have lesions [17]. The use of both elbows in the diagnostic process can also help veterinarians detect subtle differences which may have not been detected if only one of the elbows were inspected [4]. Although this is done by a trained veterinary radiologist, the diagnosis can be subjective due to variations in symptom presentation and imaging interpretation [4]. Early detection is crucial for managing symptoms and disease progression.

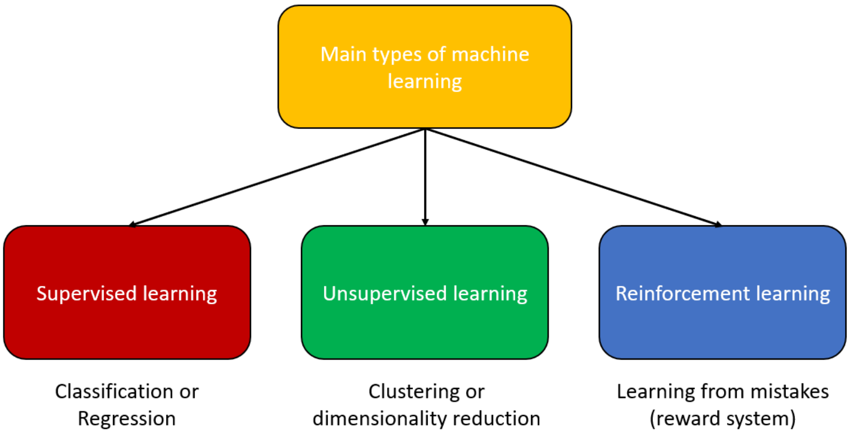
Routine screening for elbow dysplasia is recommended for breeds prone to joint diseases. Depending on the dog breed, dog population and screening methods, ED may occur on between 0% and 55% of the dog population [1]. Some examples include prevalence in 70% of Bernese Mountain dogs in The Netherlands and 17% in Labrador Retrievers in the UK [18]. In Norway, routine screening for ED is typically scheduled for when the patient is minimum 12 months old [19]. There are five breeds required to undergo screening for ED in Norway. These are: Bernese Mountain Dog, Labrador Retriever, St. Bernard, White Swiss Shepherd Dog and the Newfoundland [3].

Screening results are important in breeding programs. Dogs diagnosed with any grade of ED are generally advised against breeding to prevent the propagation of the traits associated with ED [6]. This practice is crucial in efforts to reduce the prevalence of the disease in future generations [6]. Several breed clubs and registries such as NKK, mandate such screenings for breeding animals and often publish the results to maintain transparency and guide breeding decisions within the community [3].

2.3 Machine learning

Machine learning (ML) is a subset of artificial intelligence (AI) that enables computers to learn from and make decisions based on data, rather than following explicitly programmed instructions. Its roots can be traced back to the mid-20th century, with the seminal work of Arthur Samuel in 1959, who coined the term "machine learning" while working on a checkers-playing program [20]. Over the decades, ML has evolved significantly, leveraging statistical, probabilistic, and optimization techniques to improve algorithms' predictive accuracy. Today, it is applied across various domains, including image and speech recognition, natural language processing, healthcare, finance, and environmental science, where it aids in pattern recognition, predictive modeling, and decision-making processes, thereby transforming data into actionable knowledge [21].

2.3.1 Types of machine learning



Main types of machine learning
Supervised learning
Unsupervised learning
Reinforcement learning
Classification or Regression
Clustering or dimensionality reduction
Learning from mistakes (reward system)

Figure 2.2: Three main types of machine learning: Supervised Learning, Unsupervised Learning and Reinforcement Learning.

As shown in Figure 2.2, there are three main types of machine learning: Supervised, unsupervised and reinforcement learning. Each of these have their own goals and uses [21].

Supervised learning

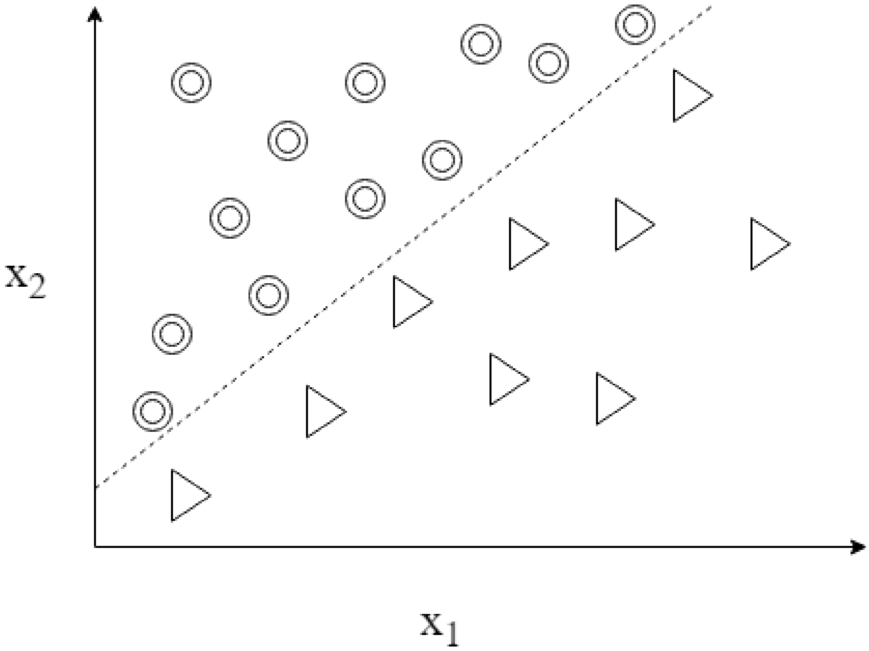


Figure 2.3: Scatter plot demonstrating the classification of two distinct groups, represented by circles and triangles. The dashed line serves as the decision boundary that separates the two categories based on the values of variables X_1 and X_2. The boundary illustrates the model's ability to distinguish between the two classes based on the input features. Based on Raschka and Mirjalili [21]

Supervised learning trains models on labeled data to perform tasks like classification and regression. In classification, the goal is to categorize inputs into classes, such as identifying cats and dogs from measurements [21]. A simple example of binary classification is illustrated in figure 2.3. Regression aims to predict numerical values, such as estimating Oslo's housing prices over time. An illustration of a linear regression task like this, can be seen in figure 2.4. These core supervised learning tasks enable precise models to be made for a wide range of predictive applications.

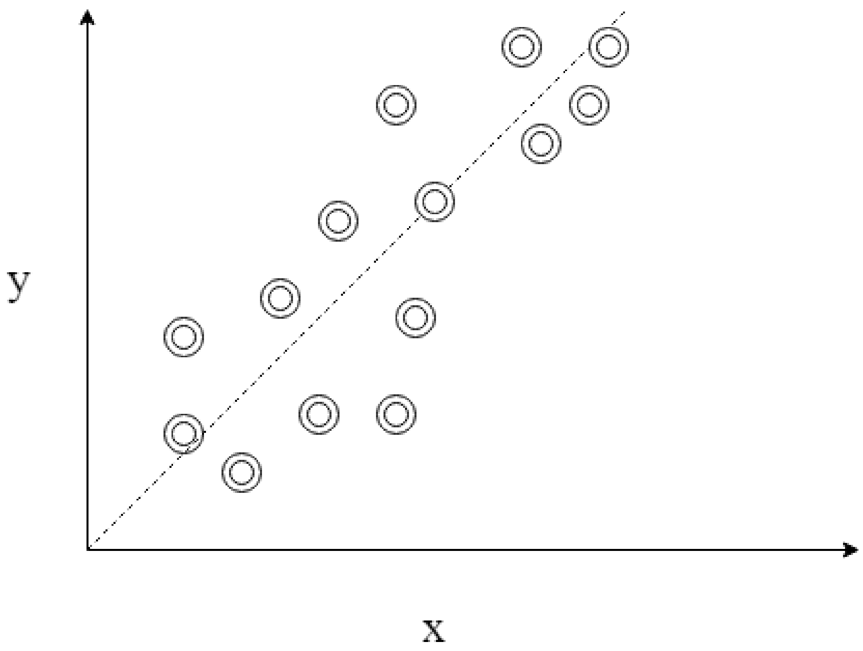


Figure 2.4: Scatter plot displaying the relationship between variables X and Y. The circles represent individual data points, and the dashed line represents the linear regression line fitted to the data, which indicates the trend and direction of the relationship between the variables. Based on Raschka and Mirjalili [21]

Multiclass classification extends binary classification to scenarios where inputs must be categorized into one of three or more classes [21]. For example, distinguishing among multiple animal species based on their measurements falls under this category. Such an example is illustrated in figure 2.5. As machine learning models require numerical input, categorical features are often transformed using techniques such as one-hot encoding [21]. In this process, a categorical feature like "color" with classes red, green, and blue would be encoded into a binary matrix.

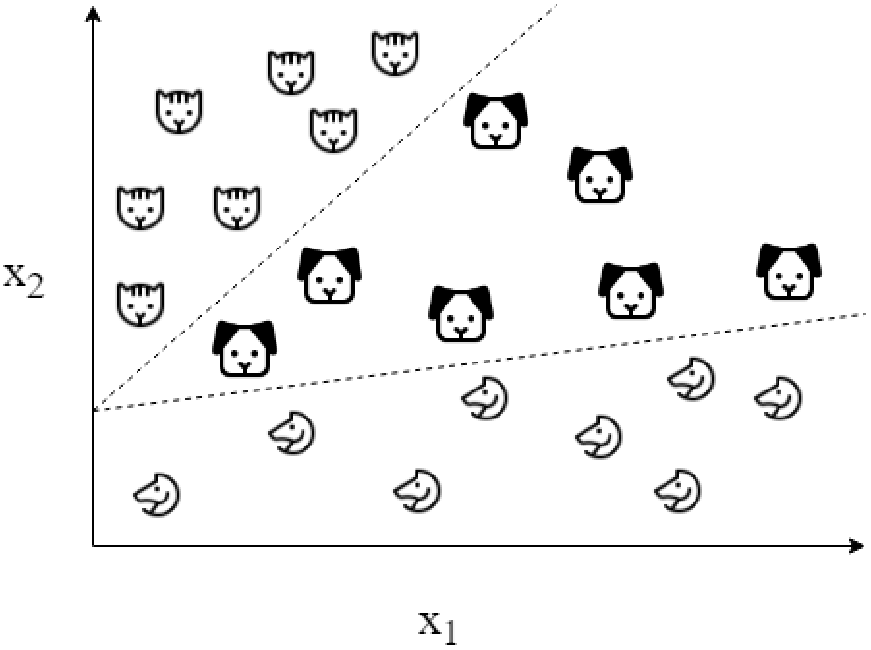


Figure 2.5: Scatter plot illustrating the classification of three distinct classes: cats, dogs, and horses. Each class is represented by a unique symbol. The dashed lines represent decision boundaries that delineate the three categories based on the values of variables X_1 and X_2. These boundaries demonstrate the model's capability to differentiate among the classes using the input features. Adapted from Raschka and Mirjalili [21]

In supervised learning, model development involves three distinct phases; training, validation, and testing. Initially, the model is trained on a specific dataset, referred to as the training set [21]. Following training, the model's performance is evaluated on the validation set, a subset of data not previously encountered by the model, to adjust hyperparameters and prevent overfitting. Finally, the model is assessed on the test set to gauge its generalized performance on unseen data [21]. Typically, the data split might involve around 50% of the data for training, 25% for validation, and 25% for testing, although these proportions can vary based on the dataset size and specific requirements of the project. Figure 2.6 visualizes the split for the train, validation and test sets.

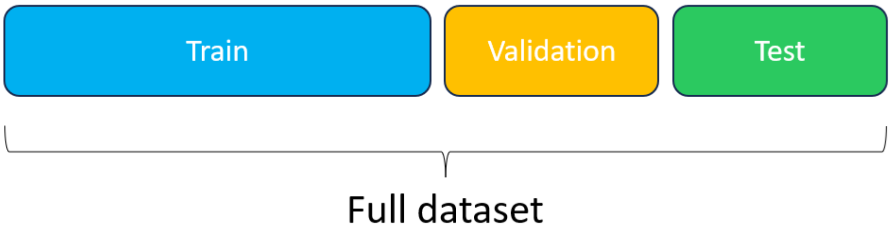


Figure 2.6: Representation of data splitting in machine learning. The full dataset is divided into three subsets: the training set, the validation set, and the test set. Typically, the training set comprises approximately 50% of the full dataset, with the validation and test sets each accounting for about 25%.

Unsupervised learning

Unsupervised learning is used when there are no labels, but rather to find "hidden" patterns in data [21]. Some examples include finding clusters of similar data points, reducing dimensionality of the data or detect noise or outliers. These methods are often used before continuing with supervised learning methods.

Some of these methods are *K-means clustering*, *Principal Component Analysis (PCA)* and *autoencoders* [21]. K-means clustering is a widely used method for grouping data into a predefined number of clusters based on feature similarities, while PCA is used for dimensionality reduction while preserving as much variance as possible [21]. Lastly, autoencoders are neural networks designed to replicate their inputs at their outputs, effectively learning data codings in an unsupervised manner. These codings can be used for noise reduction or feature extraction, which can then be used to improve the performance of supervised learning models [22].

Reinforcement learning

Reinforcement learning trains algorithms through a system of rewards and penalties, simulating a learning environment akin to natural learning behaviors [21]. It's pivotal in areas where sequential decision-making is important, such as game playing, robotics, and navigation.

Some important algorithms in reinforcement learning, are *Q-learning* and *Deep Q-Networks*. Q-learning is where an agent learns a policy to act optimally by learning the expected utility of an action taken in a given state [21]. This method is particularly effective as it does not require a model of the environment, enabling it to be applied in a variety of real-world scenarios where the dynamics are unknown [21]. Deep Q-Networks, also sometimes referred to as Deep Fitted Q Iteration, combine Q-learning with deep neural networks, extending this approach to problems with high-dimensional state spaces, such as video games or robotic control [23]. Deep Q-Networks enhance traditional Q-learning by using deep neural networks to approximate the Q-value functions, which significantly improves learning stability and efficiency in complex environments [23].

2.3.2 Performance metrics

In machine learning, evaluating model performance accurately is crucial [21]. Metrics such as Mean Squared Error (MSE), accuracy, precision, specificity, recall, F1 score, and the Matthews Correlation Coefficient (MCC) provide insights into various aspects of model predictions [24]. MSE assesses prediction error magnitudes in regression, while accuracy, precision, specificity, recall, and the F1 score address classification tasks, balancing the detection of positive instances against the backdrop of potential false positives or negatives. MCC offers a comprehensive evaluation in binary classification contexts, factoring in all quadrants of the confusion matrix [25]. These metrics collectively facilitate a nuanced understanding and comparison of model performance, ensuring models are both accurate and relevant to the task at hand.

Mean squared error

MSE serves as a common metric for assessing the accuracy of a model's predictions. It calculates the mean of the squared discrepancies between predicted and true values [26]. This metric is especially useful in regression analyses, because it quantitatively reflects the proximity of the predicted values to the actual observations within the dataset [26]. The formula to determine MSE is described as follows

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{2.1}$$

where:

- n is the number of observations,
- Y_i represents the actual observed values,
- $\hat{Y}_i$ denotes the predicted values.

A lower MSE value indicates a model that accurately predicts the observed data, while a higher MSE signifies discrepancies between the predicted and actual values, highlighting areas where the model may require improvement [26].

Confusion matrix

A confusion matrix is a table layout that allows visualization of the performance of an algorithm, typically a classification model [21]. It is particularly useful for summarizing the performance of a model on a dataset for which the true values are known. The matrix itself is composed of two dimensions: the actual class labels and the predicted class labels, each split into the categories “Positive” and “Negative.” In a 2x2 confusion matrix, the four elements represent True Positives (TP), False Positives (FP), True Negatives (TN), and False Negatives (FN).

To interpret a confusion matrix:

- **True Positives (TP)**: Cases where the model correctly predicted the positive class.
- **True Negatives (TN)**: Cases in which the model correctly predicted the negative class.
- **False Positives (FP)**: Occur when the model incorrectly predicts the positive class when it is actually negative.
- **False Negatives (FN)**: Occur when the model incorrectly predicts the negative class when it is actually positive.

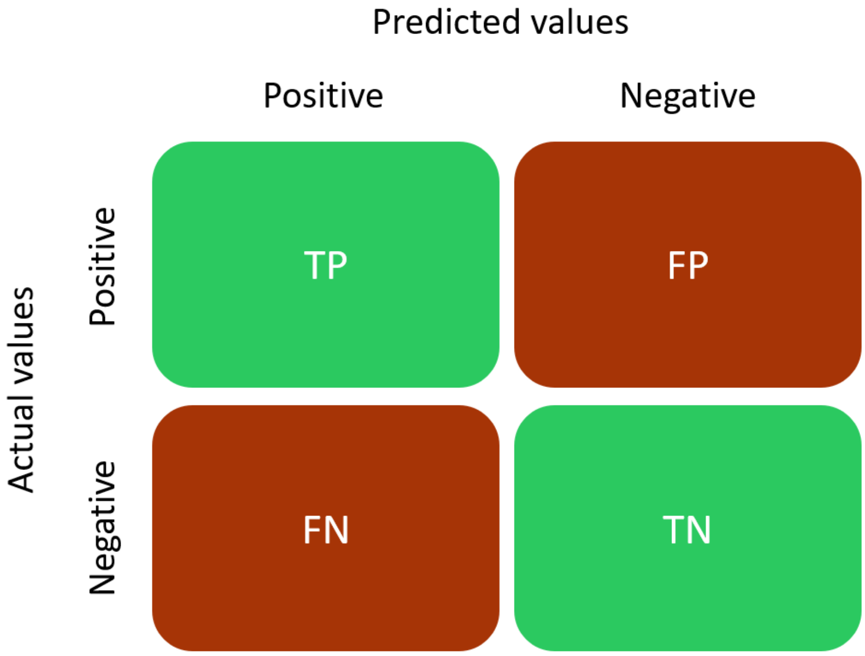


Figure 2.7: Illustration of a 2x2 Confusion Matrix used in evaluating classification model performance. The matrix segments predictions into four categories: True Positives (TP) and True Negatives (TN) represent accurate model predictions for the positive and negative classes, respectively. False Positives (FP) and False Negatives (FN) indicate errors where the model has incorrectly predicted positive and negative outcomes, respectively. Inspired by Raschka and Mirjalili [21].

An example of a basic confusion matrix can be seen in figure 2.7. The confusion matrix provides a foundation for calculating various performance metrics, such as accuracy, precision, specificity, recall, F1-score and MCC. Each of which provide different insights into the strengths and weaknesses of a model [21].

Micro and macro-averaging

In machine learning evaluation, particularly for multi-class classification, distinguishing between micro and macro-averaging methods is important to address the nuances of model performance, especially in contexts with class imbalance [21].

Micro-averaging aggregates the contributions from all classes to calculate metrics globally. For instance, micro sensitivity sums up all true positives and false negatives across the classes before calculating the total ratio [21]. This approach adjusts well to class imbalance by weighting each instance equally, highlighting the model's performance on frequently occurring classes.

Macro-averaging, in contrast, computes metrics for each class separately and then averages them, giving equal weight to each class regardless of its frequency [21]. Macro sensitivity, therefore, reflects the average effectiveness across classes without considering class imbalance. This method is particularly valuable when the impact of performance on less frequent classes is as significant as on more common ones.

Both averaging techniques provide distinct insights into model accuracy and are chosen based on the specific requirements of the task and the critical aspects of performance for the application.

Accuracy

Accuracy measures the proportion of correct predictions in all predictions made by a model, encompassing both positive and negative classes [24]. It is calculated as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.2}$$

In this equation, TP stands for true positives, TN for true negatives, FP for false positives, and FN for false negatives. A model achieving high accuracy is adept at differentiating between classes accurately.

Precision

Precision quantifies the accuracy of positive predictions in classification tasks [24]. It is the ratio of true positive predictions to the total predicted positives, encompassing both true positives and false positives. The formula for precision is:

$$Precision = \frac{TP}{TP + FP} \tag{2.3}$$

High precision indicates a low rate of false positive predictions, important in scenarios where the cost of false positives is significant. Medicine is a great example, because false positive diagnoses can lead to unnecessary anxiety, stress, and even harmful treatments for patients who do not actually have a disease.

Specificity

Specificity measures the model's ability to correctly identify all negative instances. It is an important metric alongside recall in classification tasks [24]. Specificity is defined as the ratio of true negatives to the total actual negatives, which includes both true negatives and false positives. The formula for specificity is:

$$Specificity = \frac{TN}{TN + FP} \tag{2.4}$$

High specificity indicates that the model effectively identifies negative instances, which is vital in scenarios where falsely identifying a negative instance as positive (a false positive) can have significant consequences. For example, in the legal system, incorrectly identifying an individual as a suspect could unjustly affect their life.

Recall

Recall, also known as sensitivity, measures a model's ability to correctly identify all relevant instances, which include all actual positive cases [24]. It is defined as the ratio of true positives to the actual total positives, including both true positives and false negatives. The formula for recall is:

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

High recall signifies that the model effectively identifies positive instances, essential in fields where missing a positive instance has grave consequences [21]. An example of this is in medicine, where missing a true case of cancer (a false negative) can lead to delayed treatment and decreased chances of survival.

F1 score

The F1 score is the harmonic mean of precision and recall, offering a balance between them. It is particularly useful when the class distribution is uneven [25]. The formula for the F1 Score is:

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{2.6}$$

This metric combines the perspectives of both false positives and false negatives into a single measure, making it ideal for scenarios where both precision and recall are important, such as in cancer diagnosis.

Matthews correlation coefficient

Matthews correlation coefficient, abbreviated as MCC, is a robust metric for classification evaluation, measuring the correlation between observed and predicted classifications [25]. Unlike simpler metrics, MCC considers true and false positives, negatives, and its value ranges from -1 to 1, where 1 indicates perfect prediction, 0 no better than random guessing, and -1 perfect disagreement [25]. The MCC is defined as:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}} \tag{2.7}$$

This makes MCC particularly useful in evaluating models on imbalanced datasets, providing a more nuanced assessment than accuracy alone.

ROC and AUC

The Receiver Operating Characteristic Curve, often shortened to ROC is a metric used in machine learning that depicts the performance of a classification model at all classification thresholds [27]. The curve plots the two parameters true positive rate (recall) and false positive rate (1 - specificity), showing the trade-offs between them.

A perfect performing model in the context of the ROC curve is represented by a curve that passes through the upper left corner of the plot, where the true positive rate is 1 (or 100%) and the false positive rate is 0 (or 0%) [27]. This implies that the model correctly classifies all positive instances as positive (no false negatives) and all negative instances as negative (no false positives). Visually, this would appear as a curve that sharply ascends from the origin to the top-left corner and then moves horizontally across the top of the graph space, as shown by the solid green line labeled "Perfect Performance" in figure 2.8.

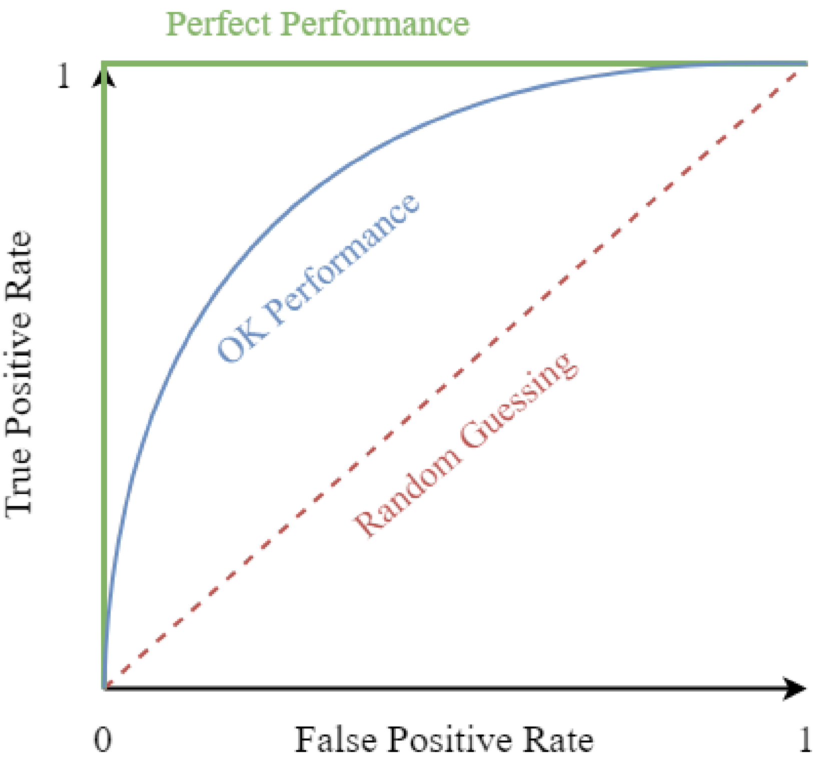


Figure 2.8: Illustration of ROC curves, with the green curve representing perfect performance, the blue curve showing OK performance, while the red dashed line depicting random guessing. Inspired by Raschka and Mirjalili [21].

The **Area Under the ROC Curve**, also known as AUC, represents a measure of the overall ability of the model to discriminate between positive and negative classes [27]. The AUC value ranges from 0 to 1, where an AUC of 0.5 suggests no discriminative ability (equivalent to random guessing, depicted by the dashed red line in figure 2.8), and an AUC of 1 indicates perfect discriminative ability, as demonstrated by the perfect performance curve. The AUC is particularly useful as it is independent of the classification threshold and provides a single measure summarizing the performance of the model across all possible thresholds. This makes it an excellent metric for comparing different models [27].

2.3.3 Artificial neural networks

Artificial neural networks (ANN) are inspired by the biological neural networks of the human brain and represent a cornerstone of machine learning. Capable of discerning intricate patterns and relationships in data, ANNs are adept at tasks including image classification, natural language processing, and decision-making [28]. They excel by iteratively refining their parameters during training, enhancing their predictive accuracy over time [29].

Neurons and architecture

In an artificial neural network, the artificial neuron serves as the core computational element [22]. It aggregates multiple inputs through a weighted sum, simulating synaptic efficacy, then processes this net input with a nonlinear activation function to produce an output [22]. This mechanism is akin to the biological neuron's response to stimuli. The architecture of ANNs arranges neurons into layers: the input layer for receiving data, hidden layers for processing, and the output layer for final predictions [29]. Information flows forward from one layer to the next, with the activation function modulating signal transmission based on a defined threshold, ensuring only meaningful signals contribute to the network's decision-making process [28].

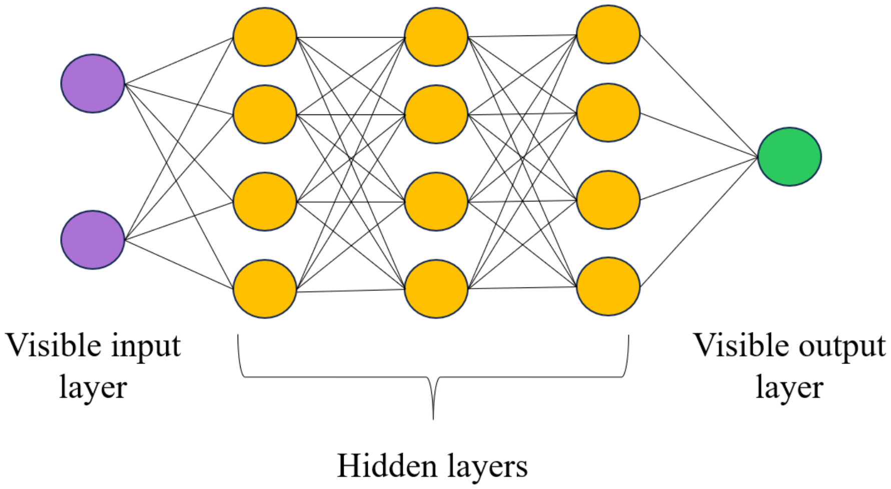


Figure 2.9: Example of a multi-layer neural network architecture for a binary problem, illustrating the arrangement of neurons into layers and the flow of information from the input to the output layer. Adapted from Chollet [22].

The structure of ANNs is organized into layers: an input layer that receives the data, several hidden layers that process the data, and an output layer that produces the final predictions (see figure 2.9) [22]. Each layer consists of multiple interconnected neurons that propagate information forward from the input to the output. The activation function in each neuron ensures that the neural network can capture non-linear relationships by modulating the signal strength as it passes through, based on a predefined threshold [28].

Figure 2.10 illustrates a single neuron model known as Adaline, which forms the basic building block for more complex neural networks [21]. By interconnecting several such units, where each serves as a node in the larger network, complex architectures like the one shown in figure 2.9 can be constructed [22]. Note that Adaline uses a linear activation function [21], which differs from the non-linear activation functions typically used in ANNs to learn complex problems. These networks are capable of learning from vast amounts of data and solving various sophisticated computational tasks, such as processing real time data from autonomous vehicles or assisting medical personnel in diagnosing diseases from images or tabular data [22].

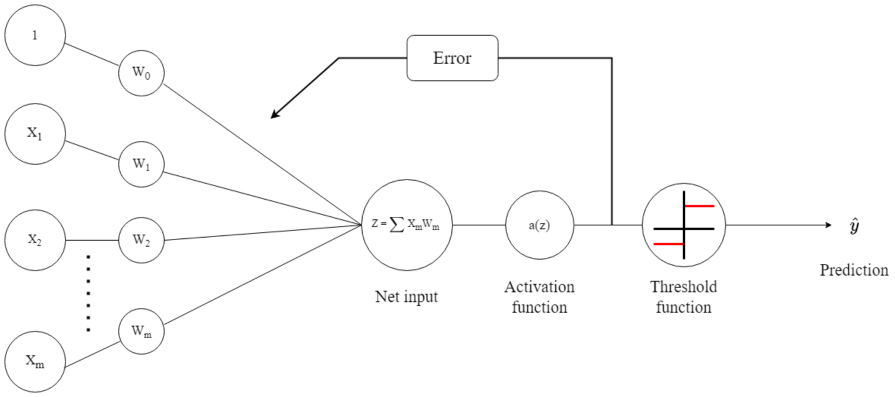


Figure 2.10: An illustration of the Adaline model architecture. The model inputs, $x_1, x_2, \ldots, x_m$, are each weighted by corresponding weights $w_1, w_2, \ldots, w_m$, and summed with a bias w_0 to form the net input z. This input is processed through an activation function $a(z)$, and the output is then passed through a threshold function to produce the final prediction $\hat{y}$. The model continuously adjusts the weights based on the error feedback to minimize prediction errors. This error adjustment is based on the difference between the prediction $\hat{y}$ and the true value y. Adapted from Raschka and Mirjalili [21].

Activation functions play an important role in neural networks, since they determine the output of a model from given inputs and help the networks recognize and learn complex patterns [22]. Activation functions introduce necessary non-linearity into the system, without which the network could not perform beyond simple linear tasks [22]. Among the various activation functions available, the Rectified Linear Unit (ReLU) and Softmax are particularly noteworthy due to their distinct roles and widespread use [22].

ReLU is a widely utilized activation function in neural networks. It functions by retaining only positive inputs and setting negative values to zero [21]. This operation is defined by the function:

$$f(x) = \max(0, x) \tag{2.8}$$

Figure 2.11 provides a graphical representation of the ReLU function. The simplicity of ReLU contributes to its computational efficiency and helps speed up the learning process in neural networks [21]. Nonetheless, a notable limitation is the phenomenon of "dying neurons," where neurons outputting zero fail to adapt further during training [22].

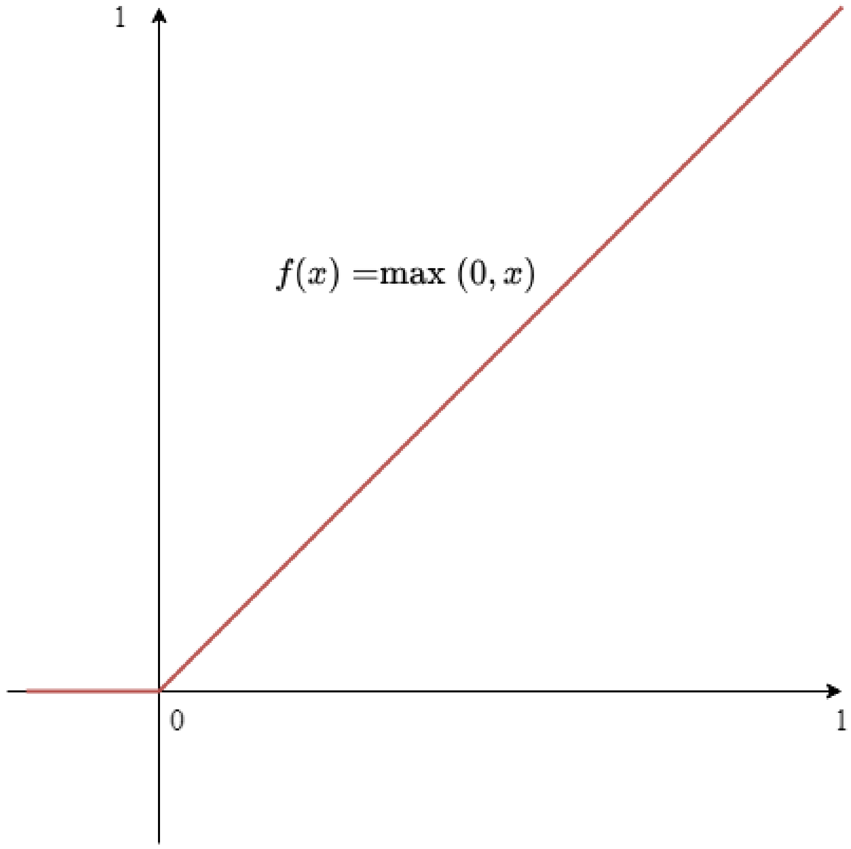


Figure 2.11: A graphical depiction of the ReLU function, showing negative inputs being set to zero and positive inputs remaining unchanged. Adapted from Raschka and Mirjalili [21].

Conversely, the softmax function is primarily employed in the output layers of neural networks designed for multi-class classification tasks [22]. It converts the raw output scores, or net inputs, into probabilities by applying a normalization process. The probability for each class is computed as follows:

$$\phi(z) = \frac{e^{z_i}}{\sum_{j=1}^{M} e^{z_j}} \tag{2.9}$$

where z_i denotes the net input for the i-th class, and M represents the total number of classes. The summation in the denominator, which includes the exponential of all class inputs, ensures all output values are non-negative and their sum equals one. This normalization provides a probability distribution across the classes, making the outputs directly interpretable in probabilistic terms [21]. For instance, if a neural network classifies images into three categories: dog, cat, or bird, and the softmax output for a particular image is [0.1, 0.8, 0.1], these probabilities correspond to the likelihood that the image depicts a dog, cat, or bird, respectively. Given these outputs, the model suggests that the image is most likely of a cat, as this category has the highest probability [21].

The efficiency of ANNs lies not only in their architecture but also in their ability to learn and adapt through training [22]. This is depicted in figure 2.12. During training, the network processes input data X (shown as yellow boxes) and produces predictions $\hat{y}$. These predictions are evaluated against the actual outputs y (represented by the two purple boxes) using a loss function (indicated by the red box on the top right). This loss function calculates the discrepancies between predictions and true outputs, generating a loss score that subsequently guides the optimization process [22].

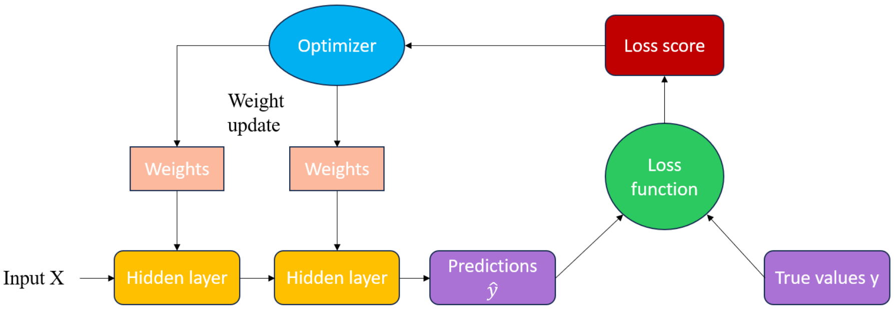


Figure 2.12: Illustration of the learning process in a neural network. The network receives input data X that passes through multiple hidden layers, transforming the input into predictions $\hat{y}$. These predictions are compared with the true values y using a loss function, which calculates the discrepancy or loss score. This score quantifies the error in the network's predictions. An optimizer then uses this loss score to update the weights of the network in a way that minimizes the loss, refining the model's accuracy over subsequent training iterations. Inspired by Chollet [22].

The optimizer (shown in blue in figure 2.12) is a component that uses the loss score to adjust the network's weights. This adjustment is used to minimize the overall error in the predictions [22]. Various types of optimizers exist, each with different strategies for weight adjustment, but all aim to refine the model's predictions by iteratively reducing the loss score. The choice of optimizer can influence the speed and quality of learning, making it an important factor in the network's training process [22].

One commonly used optimizer is the gradient descent algorithm. Gradient descent iteratively adjusts the weights of the network by moving in the direction that most reduces the loss. This is calculated as the gradient of the loss with respect to the network's weights [21]. An illustration of gradient descent can be seen in figure 2.13. Gradient descent can be further implemented in several forms, most notably stochastic gradient descent [21].

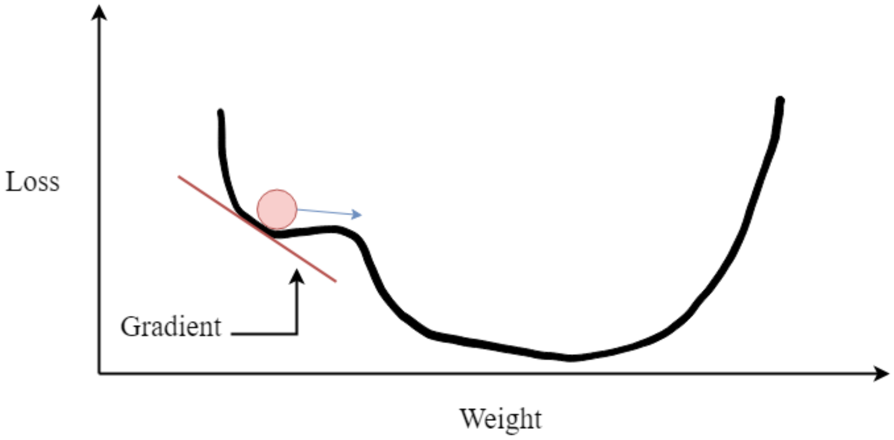


Figure 2.13: Illustration of the gradient descent optimization process. The optimizer adjusts the weights incrementally, aiming to minimize the loss function by calculating the negative gradient of the loss at each step. This process is repeated until the loss converges to a minimum, optimizing the network's performance. Adapted from Raschka and Mirjalili [21].

This learning loop from input processing to weight adjustment is vital for the neural network to improve its accuracy over time, allowing it to perform complex tasks more effectively [21]. More detailed information on the types of loss functions are covered in section 2.3.4.

Convolutional neural networks

Convolutional neural networks, or CNNs in short, are neural networks that use grid-like data such as images as input [22]. This is unlike standard neural networks which treat input data as flat vectors. CNNs preserve the spatial structure of the data, making them particularly effective for tasks such as image recognition and classification [28].

A CNN architecture typically involves several layers that transform the input image to produce an output that can be used for classification or other tasks [28]. The first layer is usually a convolutional layer, where filters are applied to the original image to create feature maps [28]. These filters detect spatial hierarchies in the data by capturing lower-level features such as edges and colors, and gradually building up to more abstract concepts through the network's depth [22]. One such example of a convolution filter is depicted in figure 2.14, where a 2x2 kernel is applied to a 3x3 input matrix to create a 2x2 feature map.

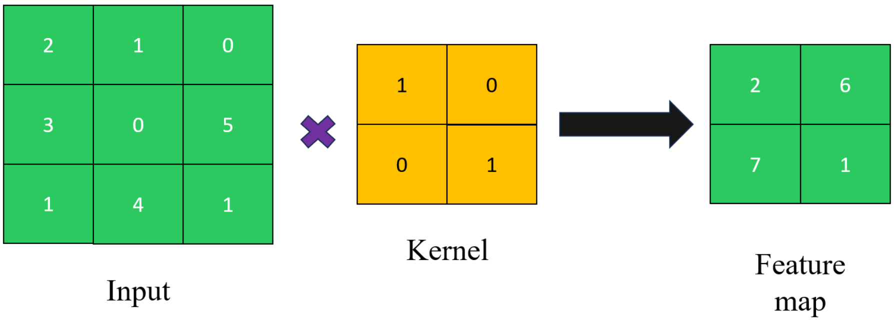


Figure 2.14: Depiction of a convolution operation in the application of CNNs. The convolution operation is applied to a 3x3 input matrix using a 2x2 kernel to produce a 2x2 feature map. The convolution process involves sliding the kernel over the input matrix, performing element-wise multiplication, and summing up the results in the corresponding position of the feature map. Adapted from Raschka and Mirjalili [21].

Following the convolutional layer, a pooling layer, such as a max-pooling layer, is often applied. This layer reduces the spatial dimensions of the feature maps, thus decreasing the computational complexity and the number of parameters [28]. Max-pooling achieves this by retaining only the maximum value in each non-overlapping sub-region of the feature map, thereby emphasizing the most prominent features while discarding less informative data [28]. This can be seen in figure 2.15, where only the maximum values of each 2x2 region of the input matrix is retained.

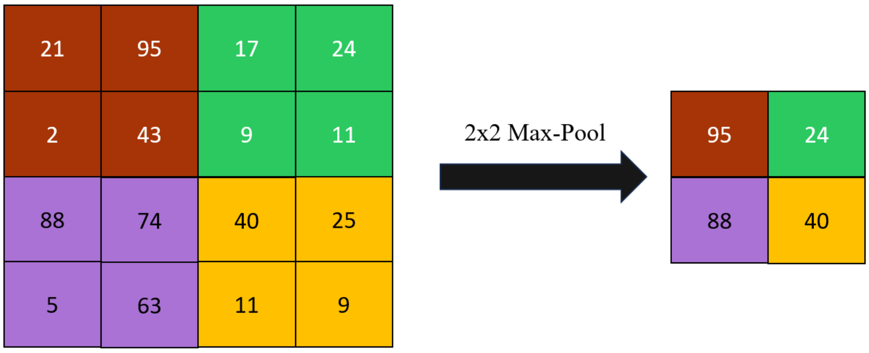


Figure 2.15: Demonstration of a 2x2 max-pooling operation applied to an 4x4 input matrix to produce a 2x2 output matrix. Each cell in the output matrix contains the maximum value of a non-overlapping 2x2 section of the input matrix, effectively reducing its spatial dimensions and retaining the most prominent features. Adapted from Stevens et al. [28].

The process of convolution and pooling is typically repeated multiple times in a CNN, each time reducing the image size and increasing the depth of feature maps to capture more complex patterns [28]. The final layers of a CNN are typically fully connected layers that use these features to classify the input image into various categories based on the training dataset [28]. Figure 2.16 depicts a sketch of a CNN used in a multi-class classification context.

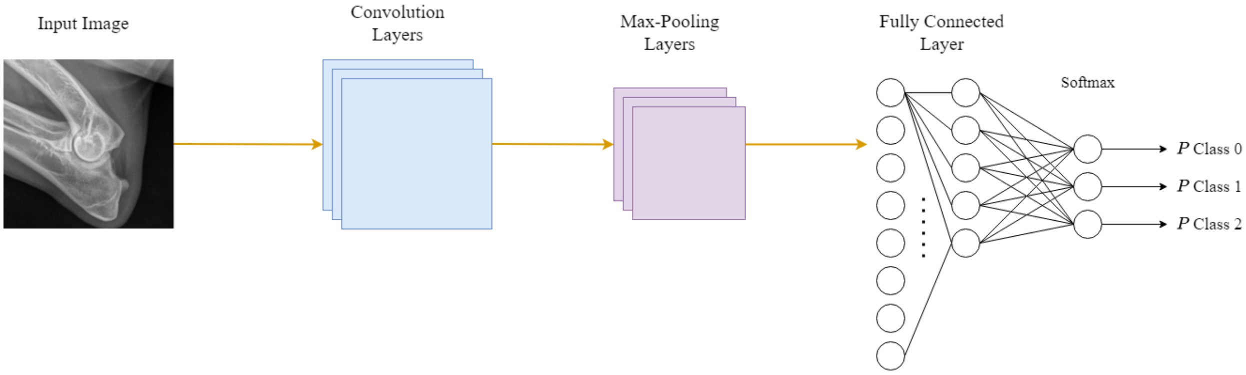


Figure 2.16: Illustration of a CNN processing an image. The network begins with an input image, in this case, an X-ray of a dog elbow. This image is first processed through several convolutional layers (blue), where features are detected at various levels of abstraction. These features are then sub sampled in the max-pooling layers (purple) to reduce dimensionality and enhance the detection of important features. The data is subsequently flattened and fed into a fully connected layer, where deeper relationships are learned. The process concludes with a softmax layer that outputs the probabilities of the image belonging to each of three classes, providing a categorical classification. Adapted from Stevens et al. [28]

Transfer learning

Transfer learning leverages the knowledge a model has gained from a previously solved, related task to improve its performance on a new, but similar problem [30]. This approach is particularly effective when the new task has a limited amount of training data available. By reusing the weights and features from a pre-trained model, such as one trained on a vast and varied dataset, the model requires less data and often less computational power to achieve high accuracy and generalization on the new task [30].

In practice, transfer learning has proven to be invaluable in fields such as computer vision and natural language processing [30]. For instance, models trained on large-scale image recognition tasks are commonly adapted to more specific and nuanced image classification tasks, like identifying specific animal species or diagnosing medical images. Similarly, models pre-trained on extensive language datasets can be fine-tuned for specific applications like sentiment analysis or language translation, significantly reducing the development time and resources required for model training from scratch [30].

EfficientNet

EfficientNet, a pretrained convolutional neural network which achieves superior accuracy and efficiency over contemporary CNNs through systematic scaling of its base model, B0 [12]. It introduces a methodical scaling strategy that scales the model dimensions; width (number of filters), depth (number of layers), and resolution (input image pixel size) optimally. The scaling of all these dimensions is called *compound* *scaling*. This enhances performance without proportional increase in computational cost [12]. An illustration of compound scaling is depicted in 2.17.

The architecture of EfficientNet is based on a baseline model, EfficientNet-B0, developed through a neural architecture search that optimizes accuracy while minimizing computational resources needed [12]. From this baseline, subsequent models (B1-B7) are scaled using the compound coefficient ϕ, which adjusts the network dimensions as follows:

$$d = \alpha^{\phi}$$
$$w = \beta^{\phi}$$
$$r = \gamma^{\phi}$$

d, w, and r represent depth, width, and resolution respectively. Constants α, β, and γ control the scaling of depth, width, and resolution with respect to ϕ.

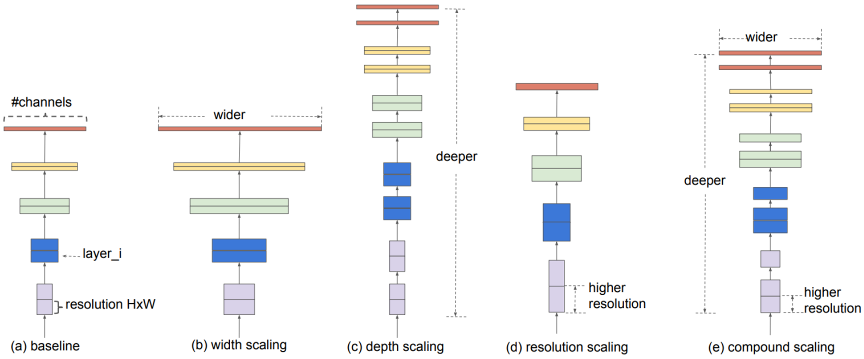


Figure 2.17: Illustration showing the scaling of models. (a) is used as reference model without scaling, while (b) - (d) show different types of scaling. (e) shows compound scaling, which is a combination of the scaling methods depicted in (b) - (d). Figure used with permission from Tan and Le [12].

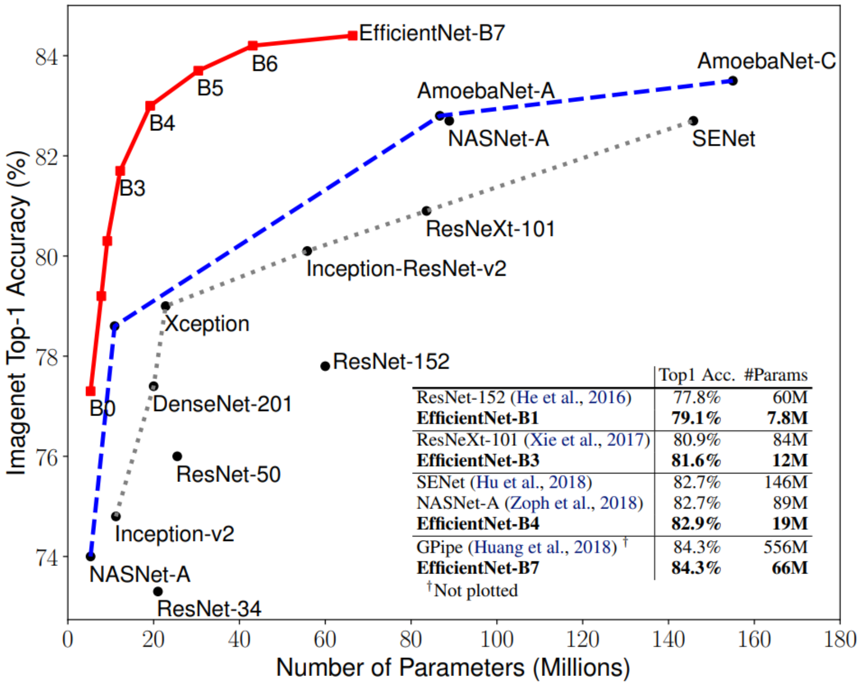
For the baseline model, EfficientNet-B0, the constants α, β, and γ were optimized under the constraint that $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$ [12]. This balance ensures that scaling one dimension does not disproportionately increase the computational load without corresponding performance gains. For EfficientNet-B0, these constants were determined to be $\alpha = 1.2$, $\beta = 1.1$, and $\gamma = 1.15$ with ϕ fixed at 1. This setup provides a structured pathway to scale the model up from B0 to B7 by varying ϕ, adapting to resource availability while maintaining a balanced increase across all dimensions [12].

Table 2.2 outlines the architectural details of EfficientNet-B0, the baseline model for the EfficientNet family. This architecture is designed through a series of stages, each employing specific operators, resolutions, channel counts, and layer repetitions [12]. The initial stage uses a simple 3x3 convolution, while subsequent stages utilize MBConv blocks with varying kernel sizes and expansion ratios to optimize both efficiency and accuracy. The progression from initial high resolution and fewer channels

Table 2.2: Table showing the architecture of the baseline model B0 in the EfficientNet family. The table shows all the stages of the model, detailing which operator is used at what stage, the resolution of the image at that time, how many channels and layers. Table reproduced with permission from Tan and Le [12].

Stage	Operator	Resolution	#Channels	#Layers
i	$\hat{\mathcal{F}}_i$	$\hat{H}_i \times \hat{W}_i$	$\hat{C}_i$	$\hat{L}_i$
1	Conv3x3	224×224	32	1
2	MBConv1, k3x3	112×112	16	1
3	MBConv6, k3x3	112×112	24	2
4	MBConv6, k5x5	56×56	40	2
5	MBConv6, k3x3	28×28	80	3
6	MBConv6, k5x5	14×14	112	3
7	MBConv6, k5x5	14×14	192	4
8	MBConv6, k3x3	7×7	320	1
9	Conv1x1 & Pooling & FC	7×7	1280	1

to lower resolution with more channels reflects a strategic design choice to balance computational cost and feature extraction capabilities [12]. This structured design achieves high efficiency, allowing subsequent models in the EfficientNet family to scale effectively across different dimensions [12].



	Top1 Acc.	#Params
ResNet-152 (He et al., 2016)	77.8%	60M
EfficientNet-B1	**79.1%**	**7.8M**
ResNeXt-101 (Xie et al., 2017)	80.9%	84M
EfficientNet-B3	**81.6%**	**12M**
SENet (Hu et al., 2018)	82.7%	146M
NASNet-A (Zoph et al., 2018)	82.7%	89M
EfficientNet-B4	**82.9%**	**19M**
GPipe (Huang et al., 2018) †	84.3%	556M
EfficientNet-B7	**84.3%**	**66M**

†Not plotted

Figure 2.18: Comparison of top-1 accuracy and number of parameters in distinct CNNs. The EfficientNet models perform better than their counterparts with equal amount of parameters. Figure used with permission from Tan and Le [12].

EfficientNet has been empirically shown to achieve state-of-the-art accuracy on ImageNet, while requiring substantially fewer parameters and lower computation than other leading CNN models [12]. This can be seen in the graph shown in figure 2.18. For instance, EfficientNet-B7 outperforms existing models like ResNet, achieving top-1 accuracy on ImageNet with significantly fewer parameters [12].

2.3.4 Loss functions

A loss function is a mathematical equation used to quantify the difference between the predicted output of a model and the actual target values, used to guide the optimization of a model's parameters [22]. The loss function's importance and relationship with the rest of the neural network is illustrated in figure 2.12, where it is depicted as the green circle.

Cross-entropy

Cross-entropy, commonly referred to as logistic loss, is a loss function used in classification tasks within neural networks [31]. This function quantifies the difference between two probability distributions; the predicted probabilities and the actual distribution represented by the true labels [31]. In machine learning, cross-entropy is primarily utilized in two forms: Binary cross-entropy or categorical cross-entropy [21].

Binary cross-entropy

Binary cross-entropy is employed in binary classification tasks, where the outcomes are limited to two classes [21]. This means that the probabilities given for random guessing will be 0.5 for each class. The equation for Binary Cross-Entropy is as follows:

$$\text{Binary Cross-Entropy} = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right] \tag{2.10}$$

In this formula, N represents the number of samples, y_i is the actual label of the i^{th} sample, and p_i is the predicted probability of the i^{th} sample belonging to one of the two classes.

Categorical cross-entropy

Categorical cross-entropy is applied in multi-class classification scenarios, where an instance can belong to one among multiple classes [22]. Its equation is an extension of the binary case and is expressed as:

$$\text{Categorical Cross-Entropy} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{M} y_{ic}\log(p_{ic}) \tag{2.11}$$

Here, M denotes the number of classes, y_{ic} is a binary indicator (0 or 1) if the class label c is the correct classification for observation i, and p_{ic} is the predicted probability that observation i belongs to class c.

Weighted kappa loss

Weighted kappa loss incorporates the kappa statistic, denoted as κ, to quantify the agreement between different raters (or classifiers) when assigning categorical labels [32]. It reflects the level of consensus among various experts when they assess the same set of cases. The kappa statistic is especially useful in ordinal classification tasks, as it accounts for the natural ordering of classes [32]. Consequently, a prediction that is closer to the true class label is penalized less severely compared to one that is farther away.

Moreover, the weighted kappa loss method employs the kappa statistic to impose a penalty within the model's loss function [32]. This penalty is directly tied to the degree of concordance between the model's predictions and the actual labels. Importantly, the penalty's intensity varies in proportion to the extent of the prediction error [32]. In essence, the more significant the deviation of the model's prediction from the true label, the greater the penalty applied. This approach serves a dual purpose: it not only incentivizes the model to enhance its accuracy but also aims to minimize the magnitude of errors in its predictions [32].

The equation for Weighted Kappa Loss is given by:

$$\text{Weighted Kappa Loss} = 1 - \frac{\sum_{i,j} w_{ij} o_{ij}}{\sum_{i,j} w_{ij} e_{ij}} \tag{2.12}$$

where o_{ij} is the observed agreement matrix, e_{ij} is the expected agreement matrix under chance, and w_{ij} is the weight matrix that quantifies the disagreement level between raters.

Chapter 3

Materials and methods

3.1 Data

The image dataset consists of 7229 x-ray images in DICOM format, sourced from various clinics across Norway and provided by Norsk Kennel Klub. It was received by veterinarians at NMBU Faculty of Veterinary Medicine in September 2022. These images, captured by multiple radiologists, were split into training, validation, and testing sets. They were classified into four classes: one class for normal elbows and three classes for abnormal elbows, with the latter categorized into three levels of increasing severity. The research group at NMBU has had access to the images and metadata about the dogs in a restricted folder. This in turn preserves the privacy of both the dogs and dog owners.

The distribution on the amount of normal and abnormal elbows in the train, validation and test sets were different from each other. For the train and validation sets, there was an upsampling of abnormal elbows to create a balanced dataset. This was done to better train the CNNs for abnormal cases and reduce overfitting [21]. On the other hand, the test set was made to best reflect the real world situation where most dogs have normal elbows [6]. This was to ensure that the model could potentially be used in real world scenarios.

3.1.1 The four-class dataset

The four-class dataset was divided into training, validation and test sets as follows:

Training set (4000 images)

The training set includes:

- 1716 images of normal (class 0) elbows (42.9%)
- Images of abnormal elbows categorized into three levels of increasing severity:
  - Class 1: 1039 images (25.98%) – including 794 images of arthrosis and/or sclerosis, 223 of arthrosis, and 22 with sclerosis.

  - Class 2: 661 images (16.5%) – including 463 images of arthrosis and 198 suspected of having MCD.
  - Class 3: 584 images (14.6%) – including 340 images of MCD, 190 of arthrosis, 43 of UAP, and 11 of OCD.

Validation set (1000 images)

The validation set consists of:

- 500 images of normal (class 0) elbows (50%)
- Images of abnormal elbows categorized into three levels of increasing severity:
  - Class 1: 197 images (19.7%) – including 153 images of arthrosis and/or sclerosis, 41 of arthrosis, and 3 with sclerosis.
  - Class 2: 150 images (15%) – including 100 images of arthrosis and 50 suspected of having MCD.
  - Class 3: 153 images (15.3%) – including 90 images of MCD, 50 of arthrosis, 10 of UAP, and 3 of OCD.

Test set (2229 images)

The test set is allocated as:

- 1983 images of normal (class 0) elbows (89.0%)
- Images of abnormal elbows categorized into three levels of increasing severity:
  - Class 1: 100 images (4.5%) – including 80 images of arthrosis and/or sclerosis and 20 of arthrosis.
  - Class 2: 75 images (3.4%) – including 60 images of arthrosis and 15 suspected of having MCD.
  - Class 3: 71 images (3.2%) – including 26 images of MCD, 23 of arthrosis, 15 of UAP, and 7 of OCD.

3.1.2 The three-class dataset

A derived three-class dataset, created in February 2024, excluded the normal class to see if the model make predictions on the severity of the elbow dysplasia among abnormal elbows. Consequently, in the three-class dataset, class 0 corresponds to level 1, class 1 to level 2, and class 2 to level 3.

Training set (2284 images)

The training set includes:

- Images of abnormal elbows categorized into three levels of increasing severity:

    - Class 0: 1039 images (45.5%) – including 794 images of arthrosis and/or sclerosis, 223 of arthrosis, and 22 with sclerosis.
    - Class 1: 661 images (28.9%) – including 463 images of arthrosis and 198 suspected of having MCD.
    - Class 2: 584 images (25.6%) – including 340 images of MCD, 190 of arthrosis, 43 of UAP, and 11 of OCD.

Validation set (500 images)

The validation set consists of:

- Images of abnormal elbows categorized into three levels of increasing severity:
    - Class 0: 197 images (39.4%) – including 153 images of arthrosis and/or sclerosis, 41 of arthrosis, and 3 with sclerosis.
    - Class 1: 150 images (30.0%) – including 100 images of arthrosis and 50 suspected of having MCD.
    - Class 2: 153 images (30.6%) – including 90 images of MCD, 50 of arthrosis, 10 of UAP, and 3 of OCD.

Test set (246 images)

The test set is allocated as:

- Images of abnormal elbows categorized into three levels of increasing severity:
    - Class 0: 100 images (40.7%) – including 80 images of arthrosis and/or sclerosis and 20 of arthrosis.
    - Class 1: 75 images (30.5%) – including 60 images of arthrosis and 15 suspected of having MCD.
    - Class 2: 71 images (28.9%) – including 26 images of MCD, 23 of arthrosis, 15 of UAP, and 7 of OCD.

3.2 Software

Python 3.7.16 was utilized alongside Deoxys, an open-source framework designed by Bao Ngoc Huynh to aid radiologists by facilitating the use of deep learning for medical imaging [33]. All files, apart from the dataset, are available on GitHub [34].

A total of 27 models were tested using the Orion High Performance Computing cluster at NMBU. Orion comprises a cluster of GPUs and CPUs in four nodes, managed using the Slurm workload manager [35]. The hardware includes three NVIDIA RTX 8000 GPUs per node. The computation time for each four-class model ranged from five to six hours, whereas each three-class model required between two and three hours.

The deployment of each model on Orion necessitated the collaborative use of three essential files listed below, which utilized the Python libraries given in table 3.1. This integration ensured robust, reproducible, and optimized execution.

1. **Configuration File (.json):** This file contained all necessary configuration parameters, including dataset parameters, augmentations, input settings, model parameters, architectural details of the model, and metrics for evaluating performance.

2. **Experiment Pipeline Script (.py):** A Python script acted as the experiment pipeline, leveraging libraries such as Deoxys and TensorFlow [36]. It defined and executed the training and evaluation processes of the model, integrated custom metrics such as MCC for scoring, and managed GPU configurations to maximize computational efficiency.

3. **Batch Submission Script (.sh):** A shell script configured the job scheduler of Orion for model runs. It specified hardware requirements like memory, CPU, and GPU allocations, and set up the environment for executing the Python script using Singularity, a containerization platform.

This structure ensured that each model's deployment was executed with precision in the high-performance computing environment provided by Orion.

Table 3.1: Python libraries utilized in this thesis.

Library	**Description**
`h5py`	Used for reading and writing HDF5 files, often for storing large datasets and model weights.
`numpy`	Remarkable library for numerical computing in Python, offering tools to handle large, multi-dimensional arrays and matrices efficiently.
`pandas`	Provides data structures and data analysis tools, ideal for manipulating numerical tables and time series.
`sklearn.metrics`	Part of the scikit-learn library, this module includes a set of functions for calculating model performance metrics.
`matplotlib.pyplot`	A plotting library for creating static, interactive, and animated visualizations in Python.
`seaborn`	A data visualization library based on matplotlib, providing a high-level interface for drawing attractive statistical graphics.
`typing`	Supports type hints, providing a way to explicitly define types of variables, functions, and classes in Python.
`deoxys`	Used to load and manage deep learning models, specifically designed for medical imaging tasks.

3.3 The workflow

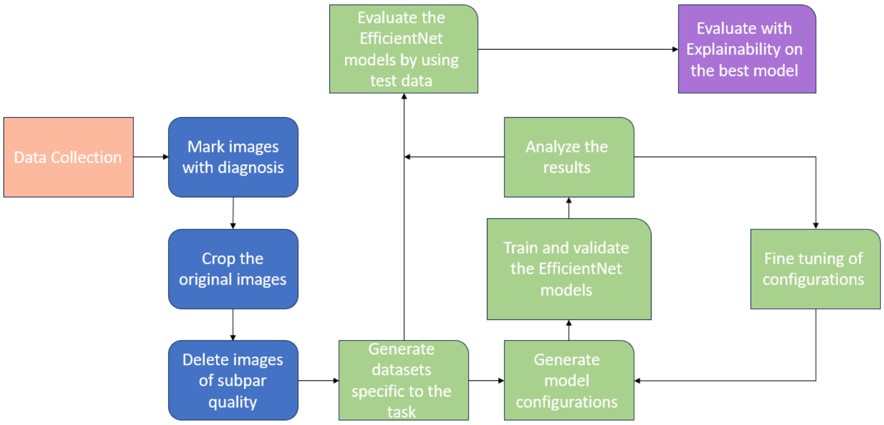

Evaluate the EfficientNet models by using test data
Evaluate with Explainability on the best model
Data Collection
Mark images with diagnosis
Analyze the results
Crop the original images
Train and validate the EfficientNet models
Fine tuning of configurations
Delete images of subpar quality
Generate datasets specific to the task
Generate model configurations

Figure 3.1: Diagram of the workflow used in this thesis (color coded). Red is raw data, blue is pre-processing, green is the dataset generation and experimentation while purple is explainability analysis.

In Figure 3.1, one can see the workflow process for this thesis. The master's student from the previous year started out by collecting data, before pre-processing the data and finally evaluating models [2]. Initially, the Norwegian Kennel Klub provided 19 878 x-ray images (the red box in the figure) to the veterinarians at NMBU's Faculty of Veterinary Medicine, a number that has increased significantly during the course of this master thesis. After pre-processing, several datasets were generated based on the problem at hand. Steiro generated datasets for binary problems and multi-class problems [2]. This was followed by the training, validation and testing phases using EfficientNet models (depicted as the green boxes in figure 3.1). The models achieving the highest accuracies were then subjected to further evaluation by testing on external datasets. Furthermore, for this thesis, there were two datasets. One with all the data, including images of both normal and abnormal elbows, and one with only images of the abnormal elbows.

3.4 Pre-processing

The four-class image dataset used in this thesis underwent complete pre-processing in September 2023. The pre-processing steps were as follows: The veterinarians marked the x-ray images with the proper diagnosis, then the images were cropped using RetinaNet, which is an object detection CNN that automatically crops images using a Focal Loss function [11]. Focal Loss takes into account the class imbalance between "background" objects and "foreground" objects in images. Afterwards, images of subpar quality were removed. For this thesis, the dataset was received fully pre-processed, requiring no further preparation before experimentation. These pre-processing steps are depicted as the blue boxes in the diagram found in figure 3.1.

A selected subset of these images was diagnosed by the veterinarians and subsequently cropped using RetinaNet. Images deemed to be of subpar quality were removed from the dataset. This part is depicted as the blue boxes in figure 3.1. As a result, the pre-processing phase concluded with 4617 images remaining for analysis during Steiro's thesis, whereas in this thesis, the number increased to 7229 images. The reason for this is that the veterinarians were continuously processing images, which were then used as external datasets for testing. For the specific research task, datasets were meticulously curated to ensure relevance and accuracy. In contrast, the three-class dataset was fully pre-processed in February 2024. The pre-processing steps for this dataset was to create a copy of the four-class dataset, then remove all images of the normal elbows. This resulted in a dataset comprised of 3030 x-ray images of abnormal elbows.

3.5 Experiment setup

For this thesis, the initial phase involved generating model configurations without the need for further data collection. This included selecting loss functions, learning rates, pre-processing types, and model complexities as detailed in table 3.2. The model configuration phase corresponds to the green segment depicted in figure 3.1. The configured models were pretrained versions of EfficientNet, each uploaded to the Orion platform for execution. Training, validation, and testing sequences followed, with the highest performing models in terms of accuracy and MCC undergoing further examination. Results were analyzed through confusion matrices and violin plots to evaluate performance. Additionally, the highest overall performing three-class model was subjected to an explainability analysis using VarGrad [37] to investigate the reasons behind its predictions. This process encompasses both the green and purple segments of figure 3.1, covering the complete experimental setup and analysis phases.

The experimentation protocol (the looped green part in figure 3.1) was structured as follows: Initial tests were conducted using a model with B1 complexity, a selected loss function, and a learning rate (LR) of 0.001. Subsequent experiments incrementally increased the complexity from B2 to B4. Following these tests, the LR was adjusted to 0.0005, and the series from B1 to B4 was repeated. Based on these results, the LR of 0.0005 consistently produced slightly higher performances than 0.001, leading to its adoption for all further testing. Regardless of the loss function or pre-processing technique employed, the complexity level of each model tested, followed a sequential progression from B1 to B4. See table 3.2 for a complete overview of the parameters and values used.

Table 3.2: Overview of parameters adjusted in the experiments.

Parameters	**Values**
Dataset	elbow_normal_abnormal_800.h5 elbow_abnormal_800.h5
Resolution	800
Complexity	B1, B2, B3, B4
Learning Rate	0.0005 0.001
Loss Functions	Categorical Cross Entropy Weighted Kappa MSE
Extra Pre-processing	No extra Three channels of images Level 1 vs 2-3 Level 1-2 vs 3

The parameters tested, given in table 3.2, were chosen to explore a range of configurations that could potentially improve the model's ability to generalize across a diverse set of x-ray images. The selection of different complexities (B1 to B4) of the EfficientNet model was aimed at investigating the trade-off between accuracy and computational efficiency. EfficientNet models are scalable in terms of depth, width, and resolution, which allows for a comprehensive evaluation of model performance across different levels of complexity [12]. This scaling is important in medical imaging tasks where increasing model depth can sometimes capture finer details in images, but may also lead to overfitting or increased computational demand [22].

The learning rate choices of 0.0005 and 0.001 were tested to determine the optimal speed at which the model learns without skipping over minima in the loss landscape, an issue often encountered when training deep neural networks [21]. The lower learning rate was anticipated to provide more stable but slower convergence, potentially leading to better generalization on unseen data [21].

Regarding the loss functions, categorical cross-entropy was chosen as a baseline loss function, because it was the only multi-class loss function used in Steiro's thesis [2]. Weighted Kappa Loss [32] and MSE [26] were selected to provide insights into the model's performance under different error sensitivities. Weighted Kappa is in theory useful in imbalanced datasets as it considers the agreement between predicted and actual classifications by weighting the different types of classification errors [32]. MSE was specifically used in a regression context to assess whether a regression approach could outperform traditional classification methods. After computing the regression results, the outputs were decoded back into categorical classes. This method was intended to explore if continuous output models offer a nuanced understanding of disease severity, which can then be translated back into discrete classes for practical diagnostic use.

The inclusion of different pre-processing techniques such as the use of three image channels, was intended to evaluate the model's ability to discern subtle features in x-rays indicative of abnormalities. In addition, binarization of the three-class dataset was also tested. This included merging two of the three classes together, making it a binary problem. Initially, experiments focused solely on testing various loss functions, learning rates, and model complexities without additional pre-processing. Upon finding the most consistent loss function, further experiments incorporated pre-processing techniques on the three-class dataset. Specifically, the three-channel approach involved using the original image in the first channel, applying histogram equalization to adjust contrast in the second [38], and enhancing contrast between neighboring pixels using an unsharp mask filter in the third channel [39], aiming to improve feature extraction and thus bolster detection capabilities.

Finally, the use of two distinct datasets, one with four classes of normal and abnormal elbows and another with three classes of only abnormal cases, was designed to test the model's diagnostic accuracy across different sample distributions. This methodological choice helps to assess the model's sensitivity and specificity.

Each of these parameter choices was systematically varied to dissect their individual and combined effects on model performance, aiming to establish a robust, accurate, and efficient diagnostic tool. This systematic testing also aids in understanding how different configurations interact with the unique characteristics of veterinary x-ray images, thereby contributing to more informed decisions in the clinical diagnosis of elbow abnormalities in dogs.

3.6 Model implementation and development

Transitioning from the initial setup to actual implementation, the next steps focused on creating and organizing the necessary files for model development. For each model, a new json file was made to accompany the python and shell script files. The json files were used to configure parameters for the CNN models, which are given in table 3.2. The files used during this thesis (excluding the json files) are shown in table 3.3. All other files can be found in github at `https://github.com/huynngoc/cubiai` [34].

Table 3.3: Overview of files used in this thesis.

Filename	**Use**	**Developer**
experiment_multiclass.py	Example code for experiment pipeline	Bao Ngoc Huynh
experiment_multiclass_linear.py	Experiment pipeline for the regression models	Bao Ngoc Huynh
experiment_multiclass_encode.py	Experiment pipeline for the binarized models	Bao Ngoc Huynh
slurm_pretrain_multiclass.sh	Example slurm script for slurm jobs in Orion	Bao Ngoc Huynh
slurm_pretrain_multiclass_linear.sh	Slurm script for slurm jobs in Orion (regression models)	Bao Ngoc Huynh
slurm_pretrain_multiclass_encode.sh	Slurm script for slurm jobs in Orion (binarized models)	Bao Ngoc Huynh
vargrad.py	Example code for vargrad	Bao Ngoc Huynh
slurm_pretrain_multiclass_test.sh	Slurm script for slurm jobs in Orion (classification models)	Bao Ngoc Huynh & Artush Mkrtchyan
experiment_multiclass_test.py	Experiment pipeline for the classification models	Bao Ngoc Huynh & Artush Mkrtchyan
explainability_test.py	Vargrad calculation adapted to the computers file locations	Bao Ngoc Huynh & Artush Mkrtchyan
customize_obj.py	File for custom objects used in this thesis, such as custom loss functions	Bao Ngoc Huynh & Artush Mkrtchyan
check_results_four_class.ipynb	Visualization of results from the four-class models	Artush Mkrtchyan
check_results_abnormal.ipynb	Visualization of results from the three-class models	Artush Mkrtchyan
check_results_regression.ipynb	Visualization of results from the regression models	Artush Mkrtchyan
csv_convert.ipynb	Conversion of .h5 to csv files to help calculate vargrad values	Artush Mkrtchyan
explainability_visualization.ipynb	Visualization of vargrad values on x-ray images	Artush Mkrtchyan
CubiAI_experiments_results.xlsx	Excel file including all the numerical metrics	Artush Mkrtchyan

There were a total of 27 model experiments in this thesis. Out of these, seven were of four-class models, while 20 were of three-class models. To keep track of the different models, a naming convention was followed. This naming convention included model complexity, learning rate, loss function and endpoints. Take for example the model *b4_0005_categorical_onehot* seen in table 3.4: *b4* explains which EfficientNet complexity the model has, *0005* means the LR is 0.0005, *categorical* explains that the loss function is categorical cross-entropy, while the endpoint *onehot* is used to explain that this model's response is one hot encoded [21]. Note that the four-class classification models all use *onehot* as endpoint.

Table 3.4: Overview of all the four-class models in this thesis.

Model complexity and LR	**Loss**	**Endpoints**	**Full model name**
b1_001	kappa	onehot	b1_001_kappa_onehot
b2_001	kappa	onehot	b2_001_kappa_onehot
b1_001	categorical	onehot	b1_001_categorical_onehot
b2_001	categorical	onehot	b2_001_categorical_onehot
b3_001	categorical	onehot	b3_001_categorical_onehot
b4_001	categorical	onehot	b4_001_categorical_onehot
b4_0005	categorical	onehot	b4_0005_categorical_onehot

An overview of the three-class models can be seen in table 3.5. All the three-class models include *level* in the name to indicate that they were made to classify only the abnormal elbows. This in turn makes them easier to distinguish from the four-class models. There are also a few models which include the endpoint *preprocess*, which is to mark that these models were trained on three channels of images. Furthermore, note that some models do not have anything written in the loss column. These are the linear and encode models. The *linear_level* models are regression models which use MSE as loss function, while the *encode_level* models are training and classifying using a binarized version of the three-class dataset. The classes in the binarized version includes level 1 vs. levels 2 and 3 merged, and levels 1 and 2 merged vs. level 3.

Table 3.5: Overview of all the three-class models in this thesis.

Model complexity and LR	**Loss**	**Endpoints**	**Full model name**
b1_001	categorical	onehot_level	b1_001_categorical_onehot_level
b2_001	categorical	onehot_level	b2_001_categorical_onehot_level
b3_001	categorical	onehot_level	b3_001_categorical_onehot_level
b4_001	categorical	onehot_level	b4_001_categorical_onehot_level
b1_0005	categorical	onehot_level	b1_0005_categorical_onehot_level
b2_0005	categorical	onehot_level	b2_0005_categorical_onehot_level
b3_0005	categorical	onehot_level	b3_0005_categorical_onehot_level
b4_0005	categorical	onehot_level	b4_0005_categorical_onehot_level
b1_0005	categorical	onehot_level_preprocess	b1_0005_categorical_onehot_level_preprocess
b2_0005	categorical	onehot_level_preprocess	b2_0005_categorical_onehot_level_preprocess
b3_0005	categorical	onehot_level_preprocess	b3_0005_categorical_onehot_level_preprocess
b4_0005	categorical	onehot_level_preprocess	b4_0005_categorical_onehot_level_preprocess
b1_0005	-	linear_level	b1_0005_categorical_linear_level
b2_0005	-	linear_level	b2_0005_categorical_linear_level
b3_0005	-	linear_level	b3_0005_categorical_linear_level
b4_0005	-	linear_level	b4_0005_categorical_linear_level
b1_0005	-	encode_level	b1_0005_categorical_encode_level
b2_0005	-	encode_level	b2_0005_categorical_encode_level
b3_0005	-	encode_level	b3_0005_categorical_encode_level
b4_0005	-	encode_level	b4_0005_categorical_encode_level

3.7 Performance metrics and evaluation

In this thesis, the following five metrics were employed to evaluate the EfficientNet models: Accuracy, MCC, AUC, Sensitivity, and Specificity. These metrics were selected to ensure a comprehensive assessment of model performance. See section 2.3.2 for more detailed descriptions.

Accuracy was chosen to measure the overall effectiveness of the models, representing the proportion of total correct predictions among all cases [24]. Given the class imbalance in the test dataset, which could skew accuracy, MCC was used as a complementary metric. MCC offers a balanced evaluation, factoring in true and false positives and negatives, and is reliable even when classes are unevenly distributed [25].

AUC was included to examine the models' discriminative ability, measuring the likelihood that a model correctly ranks a randomly chosen positive instance higher than a negative one [27]. The models provide predicted class probabilities for all labels by applying a softmax activation function to the net input at the final layer of the EfficientNet models. This process transforms the outputs into probabilities that sum to one, facilitating a direct comparison among classes [21]. AUC is then calculated based on these probabilities, ensuring an accurate assessment of the model's ability to distinguish between classes [27]. Furthermore, the final predicted class is determined by selecting the label with the highest probability from this softmax output. More details about the softmax activation function are covered in section 2.3.3.
Additionally, sensitivity and specificity were calculated for two models to facilitate comparisons with literature in section 5.2. Sensitivity assesses the proportion of actual positive cases correctly identified, essential for not overlooking any affected dogs [24]. Specificity measures the ability to identify true negatives, which is vital

to avoid unnecessary interventions for healthy dogs [24].

To deepen the analysis, confusion matrices and violin plots were also utilized. Confusion matrices provided a visual and quantitative understanding of how well the EfficientNet models performed across different classes of ED. Violin plots were used to illustrate the distribution of misclassifications and predicted probabilities for each class, shedding light on model confidence and the overlap between classes.

3.8 Explainability using VarGrad

Explainability in deep learning models is important for validating the reliability and understanding the decision-making processes of the models, especially in sensitive fields such as medical imaging [40]. This thesis employs Variance of the Gradients (VarGrad) as a method to achieve model explainability [37]. VarGrad enhances traditional gradient-based approaches by calculating the variance in the gradients of the model's output with respect to small perturbations applied to the input images. Specifically, the model's gradients were calculated when it was perturbed by normally distributed noise with a standard deviation of 0.05. This method involves slightly modifying the test images, running predictions with 20 repetitions per image, and then computing the gradients of the output with respect to the input. The variance of these gradients is then analyzed, providing insights into the regions of the image most influential in model predictions [37].

VarGrad was employed on the test images from the three-class dataset. The model chosen for deploying VarGrad had complexity B3 with a learning rate of 0.0005 without extra pre-processing. The process for calculating the VarGrad values took about six hours for the 246 x-ray images in the test set. This approach helped paint an understanding for what the model looked for when classifying the different diagnoses.

3.9 Use of AI

The use of Artificial Intelligence tools was aligned with the guidelines provided by NMBU [41]. These guidelines ensured the ethical and effective use of AI in academic work. Primarily, AI was utilized to improve LaTeX tables and refine the writing process; acting as an advanced spell checker and language editor, it assisted in improving the clarity and precision of the text. Additionally, AI served as an invaluable resource for debugging code and diagnosing programming errors. By providing explanations for errors and suggesting solutions, AI significantly streamlined the development and troubleshooting processes in the computational aspects of the research. The AI tool used specifically for improving LaTeX tables and text refinement, was a custom made ChatGPT 4 model, modified to act as an advisor, while Github Colab was used for code debugging.

Example AI prompts used:

- "I have written this about x-rays in the theory section. Please look at it and make the language easier to read, more precise and concise. No fillers!"
- "I have made this table, but it does not look good and overlaps the page. Help me fix it."
- "I don't like this part of the text. Give me some suggestions to how I can make it better."
- "Here is the error I got from the code. Help me understand the problem and come with suggestions to possible solutions."

Chapter 4

Results

4.1 Model performance

4.1.1 Performance of four-class models

Table 4.1: Overview of the performances of the four-class models, with the highest performing four-class model in terms of accuracy marked in bold.

Model	Best epoch	val acc	test acc	test MCC	test AUC
b1_001_kappa_onehot	32	0.496	0.859	0.038	0.906
b2_001_kappa_onehot	26	0.400	0.593	0.052	0.537
b1_001_categorical_onehot	38	0.811	0.927	0.699	0.982
b2_001_categorical_onehot	36	0.817	0.927	0.703	0.978
b3_001_categorical_onehot	20	0.810	0.899	0.620	0.979
b4_001_categorical_onehot	26	0.809	0.943	0.748	0.986
b4_0005_categorical_onehot	**49**	**0.845**	**0.958**	**0.805**	**0.986**

In this thesis, seven out of the 27 models were trained on the four-class dataset. The evaluation of these models considered a suite of metrics: accuracy, MCC, AUC, confusion matrices and violin plots. These measures and plots collectively guided the selection of the highest overall performing model, ensuring a choice that reflects comprehensive performance, not solely based on accuracy.

The analysis of four-class model performances reveals a significant disparity in effectiveness between models using the kappa loss function and those employing categorical cross-entropy. As shown in table 4.1, models utilizing kappa loss (e.g., *b1_001_kappa_onehot* and *b2_001_kappa_onehot*) consistently demonstrate lower validation accuracy, test accuracy, test MCC, and AUC values compared to their counterparts using categorical cross-entropy. This discrepancy suggested that while kappa loss might provide theoretical benefits for handling imbalanced datasets, it does not translate well into practical improvements in this specific application, leading to its discontinuation in further experiments.

Attached in figures 4.1 and 4.2, one can see confusion matrices which show the accuracy for the validation and test set for *b1_001_kappa_onehot*. Note that only the first and third true class label rows showed numbers above zero (class 0 and 2).

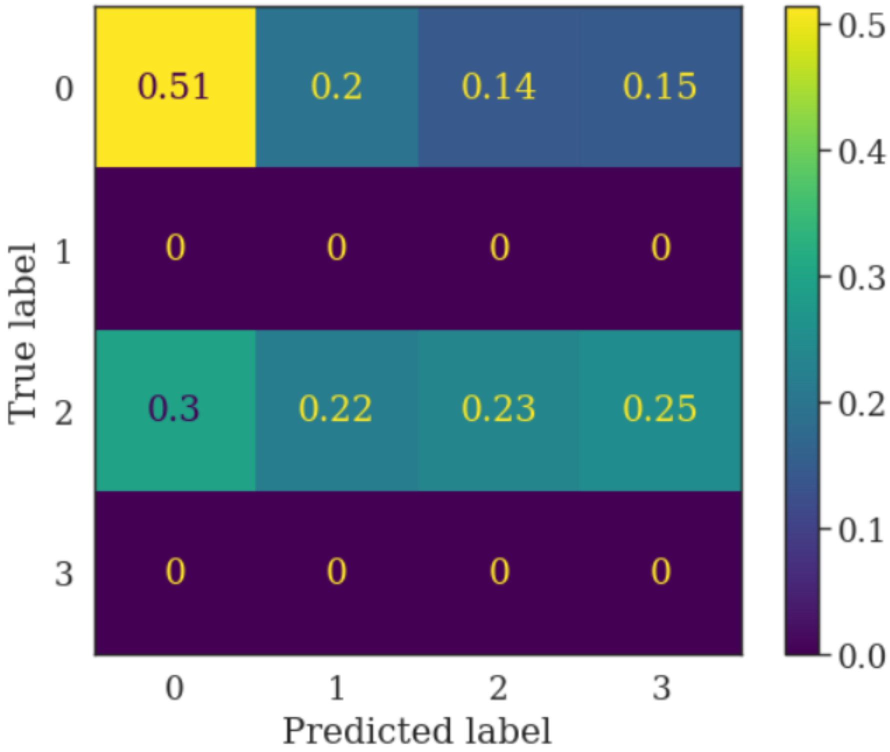


Figure 4.1: Confusion matrix of the validation set for the four-class kappa loss model with complexity B1 and learning rate 0.001.

The confusion matrix presented in figure 4.1 illustrates the model's limitations in accurately classifying all categories. Class 0 (normal elbows) had the highest correct classification rate at 51%, but experienced significant misclassifications: 20% were classified as class 1, 14% as class 2, and 15% as class 3. Neither class 1 nor class 3 had any correct predictions. Class 2, however, achieved a 23% accuracy rate, with misclassifications distributed as follows: 30% mistaken for class 0, 22% for Class 1, and 25% for Class 3.

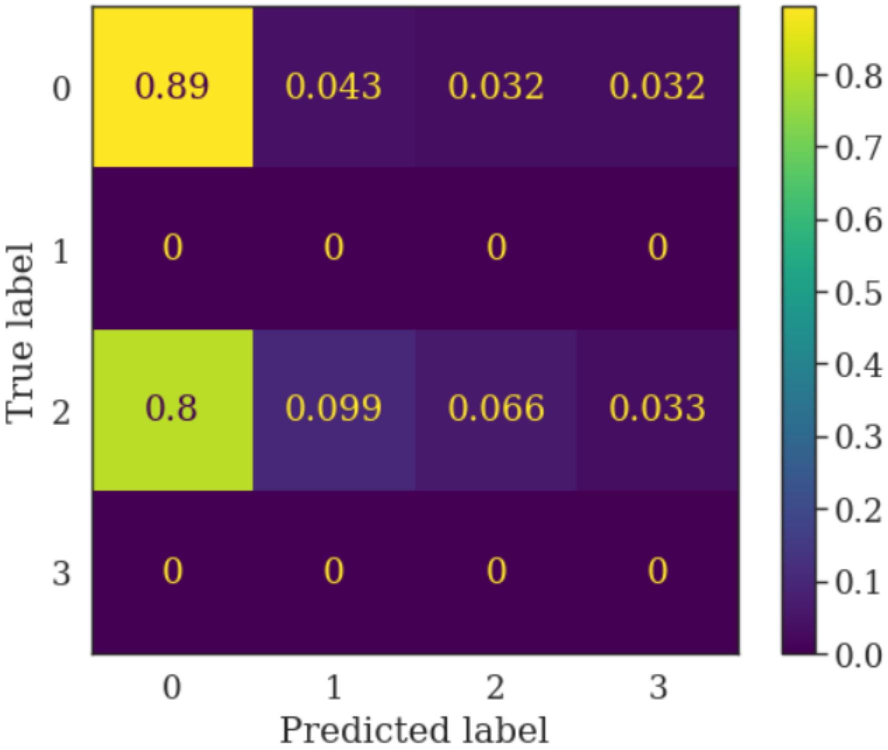


Figure 4.2: Confusion matrix of the test set for the four-class kappa loss model with complexity B1 and learning rate 0.001.

Figure 4.2 depicts the confusion matrix for the kappa loss model with complexity B1 and learning rate 0.001 for the test set. Relative to the confusion matrix for the test set in figure 4.1, one can see it boasted a higher total accuracy. For instance, it classified normal elbows (class 0) correctly 89% of the time. However, that's the only redeeming quality of this model, since there were no correct classifications of elbows with true class 1 and 3. Misclassifications of class 0 included 4.3% as class 1, and 3.2% each as class 2 and class 3.

Furthermore, the figures 4.3 and 4.4 show the confusion matrices for the validation and test set for *b2_001_kappa_onehot*. Similar to the b1 kappa model, only two of the true labels had numbers above zero. However, for this model, the true labels showing numbers larger than zero were class 0 and 1.

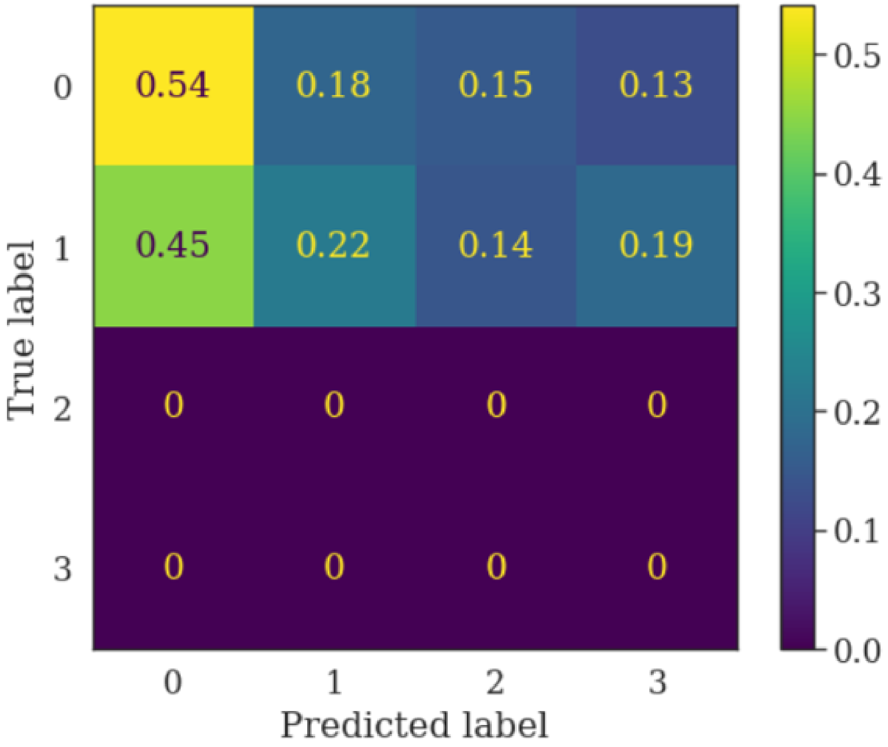


Figure 4.3: Confusion matrix of the validation set for the four-class kappa loss model with complexity B2 and learning rate 0.001.

In figure 4.3, one can see the confusion matrix for the kappa model with complexity B2 and learning rate 0.001 for the validation set. Similar to the B1 model in figure 4.1, this model struggles classifying all the different classes. The highest accuracy here is the accurate classification of class 0 with 54%, with misclassifications of this class being 18% wrongly classified as class 1, 15% as class 2 and 13% as class 3. For true class 1 on the other hand, the correct classification rate was at a meager 22%, with 45% of instances misclassified as class 0, 14% as class 2 and 19% as class 3. Lastly, there were no correct classifications of true class 2 nor 3.

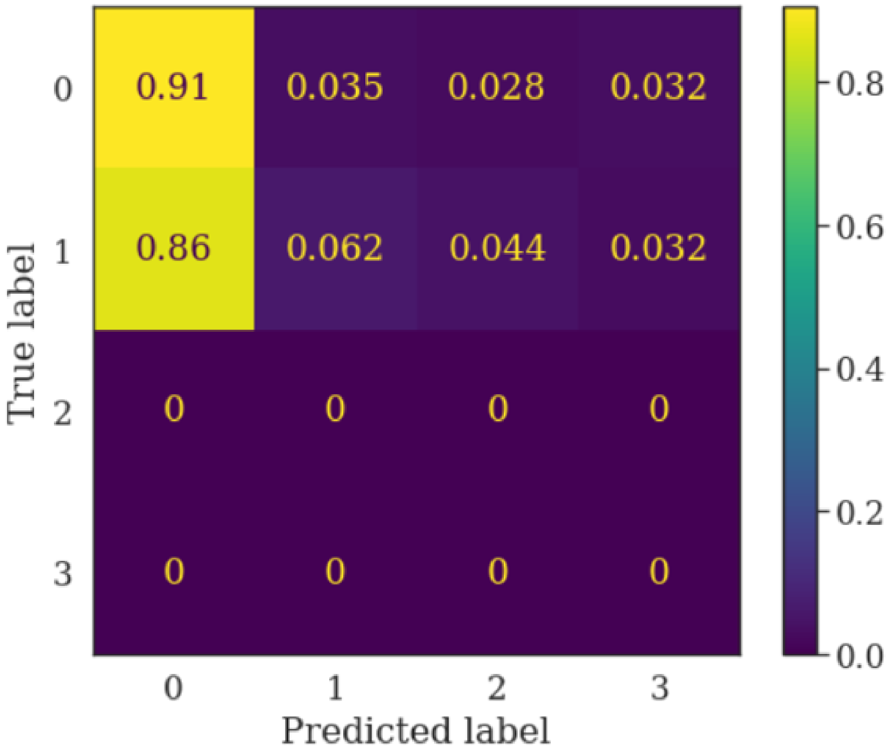


Figure 4.4: Confusion matrix of the test set for the four-class kappa loss model with complexity B2 and learning rate 0.001.

The test set's confusion matrix for the kappa model with complexity B2 and learning rate 0.001 in figure 4.4 performed similarly as its B1 counterpart seen in figure 4.2. Here the accuracy got as high as 91% for class 0, but further misclassified 3.5% of true class 0 cases as class 1, 2.8% as class 2 and 3.2% as class 3. In terms of class 1, the true classification rate was 6.2%, with 86% of true class 1 cases classified as class 0, 4.4% as class 2 and 3.2% as class 3. Similar to its validation set counterpart in figure 4.3, there were no correct instances of class 2 and class 3.

The highest overall performing four-class model was therefore the B4 model with learning rate 0.0005, utilizing categorical cross-entropy as its loss function. Its metrics are shown at the bottom of table 4.1, where it had higher metrics than the rest of the models for validation accuracy (84.5%), test accuracy (95.8%), MCC (0.805) and AUC (98.6%). Note that this was also the most accurate four-class model. Compared to the four-class model where the difference was the learning rate at 0.001, it appeared that for the four-class models, the deciding parameters for highest performance were complexity B4, learning rate 0.0005 and categorical cross-entropy as loss function.

Further, confusion matrices of the most accurate four-class model can be seen in Figure 4.5 and Figure 4.6. These figures show its performance on the validation and test data, respectively. However, it is the performance on the test data that is the most important, because of its real world relevance on how prevalent the different diagnoses are. The accompanying violin plot, depicted in Figure 4.7, further illustrates the distribution of the model's predictions across the actual classes in the test set.

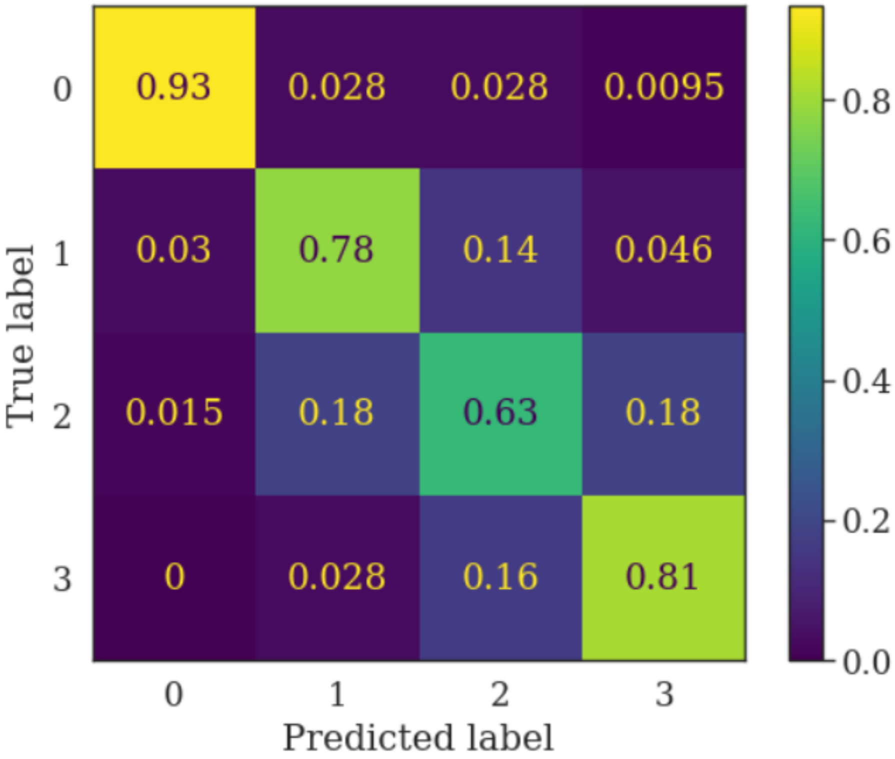


Figure 4.5: Confusion matrix of the validation set for the four-class model with the highest performance in terms of accuracy and MCC (categorical cross-entropy with complexity B4 and learning rate 0.0005).

The confusion matrix for the validation data in figure 4.5 highlights the model's high accuracy in identifying normal elbows (class 0) at 93%, and severe elbow dysplasia (class 3) at 81%. Challenges arose with classes 1 and 2, where the model exhibited a 14% misclassification of class 1 as class 2 and 18% of class 2 as class 1. Misclassifications between class 3 and class 2 occurred at a rate of 16%, and in reverse, at 18%.

Misclassifications were predominantly between neighboring classes, as shown in figure 4.5. True class 0 and class 3 images were rarely misidentified as class 1. Additionally, there were minimal instances where true class 2 was classified as class 0, and a few cases of class 3 misidentified as class 1, with no occurrences of class 3 being classified as class 0. This pattern underscored the model's tendency to confuse classes that were adjacent in the classification scheme.

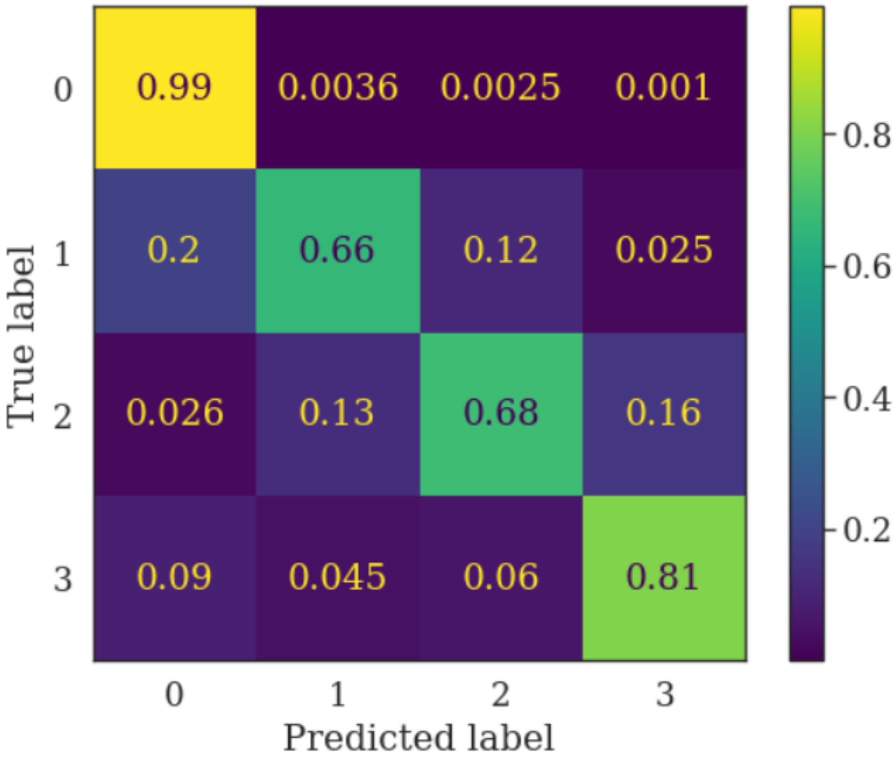


Figure 4.6: Confusion matrix of the test set for the four-class model with the highest performance in terms of accuracy and MCC (categorical cross-entropy with complexity B4 and learning rate 0.0005).

The test set's confusion matrix in figure 4.6 illustrates even higher efficacy in classifying normal elbows (class 0) with 99% accuracy, while maintaining 81% for severe dysplasia (class 3). The model particularly struggled with class 1 and 2 distinctions, misclassifying class 1 as normal in 20% of cases, and class 2 as class 1 in 12% of cases, with a reversal rate of 13%. Misclassifications of class 2 as class 3 were noted in 16% of the instances. Further, as in the validation set's confusion matrix, one can see that most misclassifications happened between adjacent classes, except for class 3.

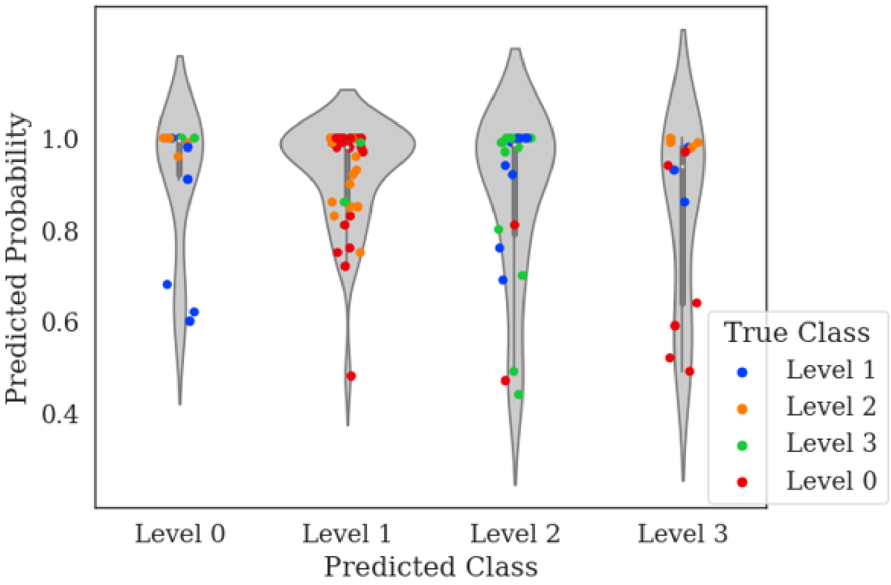


Figure 4.7: Violin plot of the test set for the four-class model with the highest performance in terms of accuracy and MCC (categorical cross-entropy with complexity B4 and learning rate 0.0005).

The violin plot in figure 4.7 illustrates the misclassifications for each diagnostic category. Each "dot" represents an individual sample, with its color denoting the true diagnostic class. The plot revealed a notable pattern: a small number of samples were misclassified into the normal category (class 0), whereas a considerable amount of truly normal samples were incorrectly predicted as class 1. Moreover, the plot indicated a tendency for samples from class 1 to be frequently misclassified as class 2, and to a lesser extent, this misclassification also affected samples from class 3. Conversely, misclassifications as level 3 were relatively uncommon.

4.1.2 Performance of three-class models

Table 4.2: Overview of the performances of the three-class models. The two models with highest test accuracy are marked in bold.

Model	Best epoch	val acc	test acc	test MCC	test AUC
b1_001_categorical_onehot_level	38	0.706	0.662	0.489	0.850
b2_001_categorical_onehot_level	41	0.726	0.674	0.521	0.835
b3_001_categorical_onehot_level	31	0.636	0.617	0.421	0.807
b4_001_categorical_onehot_level	26	0.698	0.662	0.491	0.842
b1_0005_categorical_onehot_level	41	0.722	0.723	0.579	0.861
b2_0005_categorical_onehot_level	49	0.748	0.731	0.596	0.863
b3_0005_categorical_onehot_level	**50**	**0.720**	**0.764**	**0.643**	**0.885**
b4_0005_categorical_onehot_level	38	0.742	0.731	0.593	0.872
b1_0005_categorical_onehot_level_preprocess	40	0.738	0.723	0.587	0.868
b2_0005_categorical_onehot_level_preprocess	40	0.714	0.699	0.548	0.854
b3_0005_categorical_onehot_level_preprocess	35	0.720	0.699	0.557	0.819
b4_0005_categorical_onehot_level_preprocess	39	0.736	0.719	0.578	0.852
b1_0005_linear_level	29	0.758	0.731	0.603	-
b2_0005_linear_level	38	0.764	0.743	0.618	-
b3_0005_linear_level	41	0.743	0.686	0.530	-
b4_0005_linear_level	29	0.721	0.731	0.568	-
b1_0005_encode_level	**44**	**0.730**	**0.801**	**0.699**	-
b2_0005_encode_level	33	0.736	0.715	0.567	-
b3_0005_encode_level	44	0.732	0.735	0.597	-
b4_0005_encode_level	25	0.744	0.735	0.600	-

A majority of the models in this study, accounting for 20 of the 27, focused on the three-class dataset. This approach stemmed from the four-class model's proficiency in identifying normal elbows, leading to a concentrated effort on distinguishing between the pathological classes.

In contrast to the four-class models, the performance metrics for the three-class models, as detailed in table 4.2, exhibit consistency across different configurations and complexity levels. Whether pre-processed, binarized, or standard, these models showed relatively similar accuracy, MCC, and AUC scores. This is the case for every model in the table, except for the *encode_ level* models. Here, the least complex variant (B1) performed better than the rest of the encode models in terms of test accuracy and test MCC. On the other hand, the consistent performance for the rest of the models alleviated concerns about overfitting, suggesting that the models were well-tuned to the general characteristics of the dataset rather than memorizing specific data points.

Similarly to what was noted about the four-class models, the best performing learning rate for the three-class models appeared to be 0.0005. By comparing the eight first models (*categorical_ onehot_ level*) in table 4.2, one can see that the models (5 - 8) with learning rate 0.0005 had better performance across all categories compared to the models (1 - 4) with learning rate 0.001.

The results presented in table 4.2 demonstrate that higher model complexity did not consistently lead to better performance among the three-class classification models. Specifically, the *b3_ 0005_ categorical_ onehot_ level* model recorded a test accuracy of 76.4% and a test MCC of 0.643, whereas the simpler *b1_ 0005_ encode_ level* model achieved higher results with a test accuracy of 80.1% and an MCC of 0.699. Other models, including the *b1_ 0005_ categorical_ onehot_ level_ preprocess* and *b2_ 0005_ linear_ level*, showed more modest performances with test accuracies of 72.3% and 74.3%, and MCCs of 0.587 and 0.618, respectively.

Among the *encode_ level* models, the B1 variant stood out, surpassing its nearest competitor, the B4 model, by 6.6% in test accuracy and 0.099 in MCC. However, note that the B1 variant's validation accuracy was lower than the other *encode_ level* variants'. This notable deviation suggests that the B1 model's superior performance in test accuracy and test MCC could be due to specific characteristics of the model or dataset that may not be replicated across other scenarios. This anomaly raised the possibility that the B1 model's success might be attributable to favorable conditions or model-specific advantages rather than generalizable improvements.

For the three-class models, the term *level* will be used for the classes to avoid confusion. Because, without the normal class, class 0 corresponds to abnormal elbows of level 1, class 1 to level 2 and class 2 to level 3 (as mentioned in section 3.1.2).

Furthermore, note that the test accuracy for the encode models in table 4.2 shows their overall test accuracy, while the table presented in table 4.3 shows the test accuracy for level 1 vs level 2-3 and level 1-2 vs level 3 for each encode model, respectively.

Table 4.3: Comparison of test accuracy for level 1 vs level 2-3 and level 1-2 vs level 3 across different complexity levels for the encode models.

Model	test acc Level 1 vs Level 2-3	test acc Level 1-2 vs Level 3
b1_0005_encode_level	0.87	0.90
b2_0005_encode_level	0.82	0.86
b3_0005_encode_level	0.82	0.97
b4_0005_encode_level	0.83	0.89

Looking at table 4.3, one can see that all the models performed similarly with high accuracy, with the lowest accuracy at 82% and highest at 97%. The models also tended to find it easier to classify level 3 as its own class, while still having high accuracy for level 1 as its own class.

This analysis underscores that higher complexity did not guarantee improved performance for the three-class models. The similar performance metrics of the *linear_ level* models compared to others in table 4.2 indicate that transforming the problem into a regression problem before classifying and decoding the results did not enhance performance. Refer to Appendix A for the confusion matrices of the regression models. Additionally, the lack of performance gains from utilizing images from three channels (models 9 - 12 in table 4.2) suggests that extra pre-processing did not boost performance in terms of test accuracy or test MCC.

There was therefore a conscious choice to declare *b3_ 0005_ categorical_ onehot_ level* the all-round highest performing three-class model.

The highest-performing model for the three-class challenge is represented in figure 4.8 and figure 4.9, which provide a visual assessment of validation and test data performance. To complement these, figure 4.10 visualizes the prediction spread against true diagnoses, emphasizing the model's discriminative capability on the test set.

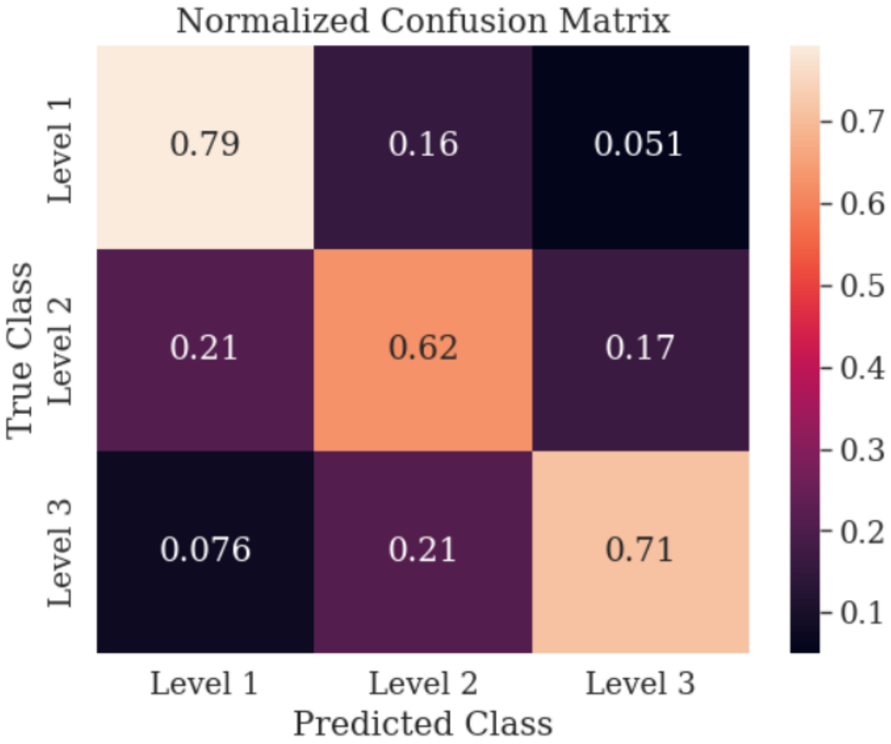


Figure 4.8: Confusion matrix of the validation set for the three-class model with the highest performance in terms of accuracy and MCC (categorical cross-entropy with complexity B3 and learning rate 0.0005).

Figure 4.8 presents the validation data confusion matrix for the top-performing three-class model. The model demonstrated a higher accuracy for level 1 at 79% and level 3 at 71%, compared to a lower 62% for level 2. Notably, level 1 instances were misclassified as level 2 in 16% of cases, and level 2 instances as level 1 in 21% of cases. Misclassifications between levels 2 and 3 were also observed, with level 2 being mistaken for level 3 in 17% of instances and vice versa in 21% of cases. Similarly to figures 4.5 and 4.6, one can see that most misclassifications happened between adjacent levels in figure 4.8.

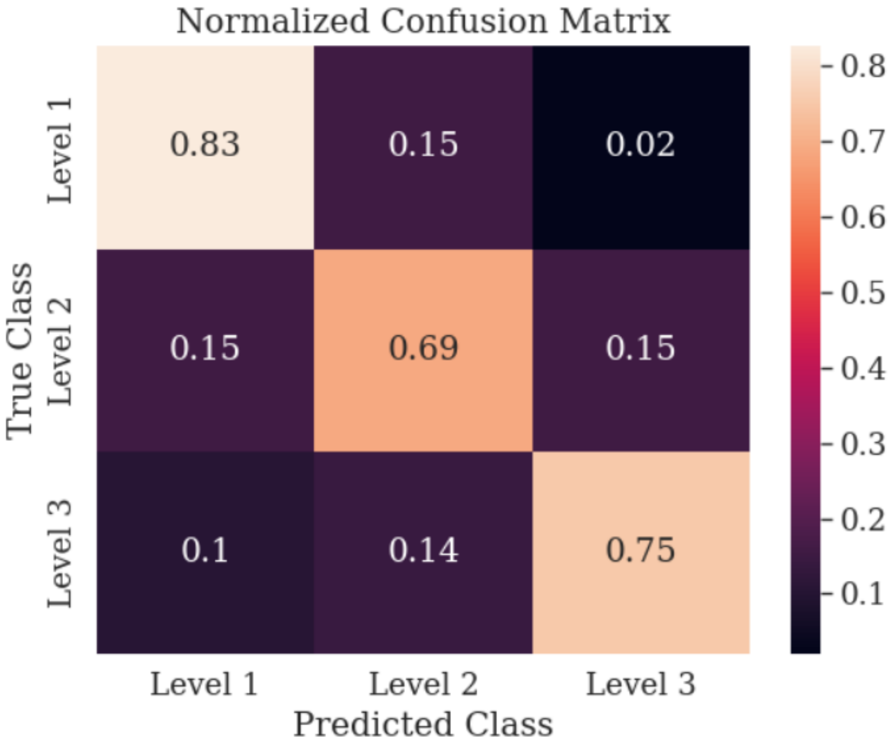


Figure 4.9: Confusion matrix of the test set for the three-class model with the highest performance in terms of accuracy and MCC (categorical cross-entropy with complexity B3 and learning rate 0.0005).

The confusion matrix of the top performing three-class model for the test set found in figure 4.9 reveals the test set accuracy for the top performing three-class model, with 83% for level 1 and 75% for level 3 diagnoses. The accuracy for level 2 diagnoses saw a slight increase to 69%, relative to the confusion matrix for the validation data in figure 4.8. The model had a balanced misclassification rate between levels 1 and 2, with each being incorrectly identified as the other in 15% of instances. Misclassification between levels 2 and 3 was 15% for level 2 classified as level 3, and 14% for level 3 classified as level 2. Given the equal misclassification rates of level 1 as level 2 and vice versa, along with the errors between levels 2 and 3, the total misclassification involving level 2 amounted to 30%. This suggests that the model found level 2 the most challenging to discriminate accurately. Again, one can see that most misclassifications happened between neighboring levels.

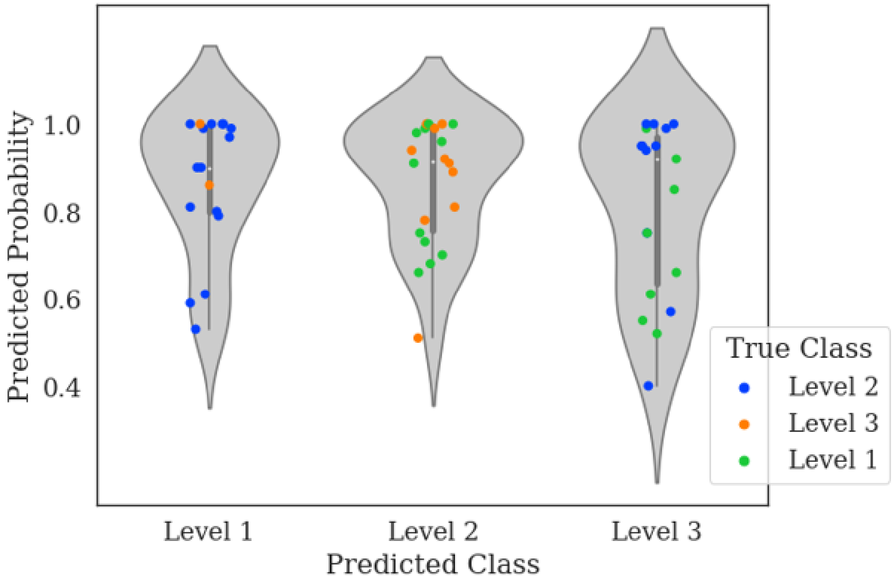


Figure 4.10: Violin plot of the test set for the three-class model with the highest performance in terms of accuracy and MCC (categorical cross-entropy with complexity B3 and learning rate 0.0005).

The violin plot in Figure 4.10 reveals the misclassification patterns in the three-class diagnostic model. True level 1 samples were seldom misclassified as level 2 or 3. A notable portion of level 2 samples were incorrectly predicted as level 1 or 3. Misclassifications of level 3 primarily occurred as level 2.

4.1.3 Overall model performance

The most accurate models from this project are depicted in table 4.4 which includes their names and main metrics. The first model in that table is a four-class model, while the rest are three-class models. As one can see, their performances were quite similar to each other with small differences in the case for the three-class models. The high test accuracy for the four-class model *b4_0005_categorical_onehot* was because 89% of the images in the four-class test set were of normal elbows (as mentioned in section 3.1.1).

Table 4.4: Overview of performances of the most accurate models. Every model which has level in its name, is a three-class model.

Model	Best epoch	val acc	test acc	test MCC	test AUC
b4_0005_categorical_onehot (four-class)	49	0.845	0.958	0.805	0.986
b3_0005_categorical_onehot_level	50	0.720	0.764	0.643	0.885
b1_0005_categorical_onehot_level_preprocess	40	0.738	0.723	0.587	0.868
b2_0005_linear_level	38	0.764	0.743	0.618	-
b1_0005_encode_level	44	0.730	0.801	0.699	-

Based on the performance metrics in table 4.4, the confusion matrices in figures 4.6 and 4.9, and the violin plots in figures 4.7 and 4.10, one can conclude that the highest performing four-class model was *b4_0005_categorical_onehot* while the highest performing three-class model was *b3_0005_categorical_onehot_level.*

Table 4.5: Summary of sensitivity and specificity metrics for the highest overall performing models. Four-Class Model corresponds to *b4_0005_categorical_onehot* while Three-Class Model corresponds to *b3_0005_categorical_onehot_level.*

Metric	Four-Class Model	Three-Class Model
Macro Sensitivity	0.7845	0.7615
Micro Sensitivity	0.9612	0.7642
Macro Specificity	0.8706	0.8700
Micro Specificity	0.8151	0.8709

A more detailed analysis of the two highest overall performing models (for their respective datasets), reveal important information about these two models' behaviors. For instance, table 4.5 reveals the sensitivity and specificity at both micro and macro levels for the highest performing three and four-class models. Macro sensitivity takes the average sensitivity across all classes, while micro sensitivity takes into account the class imbalance and computes the aggregation of all classes. The same applies for micro and macro specificity. Note that the micro sensitivity of the four-class model was 96.12%, much higher than its three-class model counterpart at 76.42%. The reason for this was that 89% of the elbows in the four-class test dataset, were of normal elbows (large class imbalance).

4.1.4 Three-class model with explainability

The model that was used with explainability was *b3_0005_categorical_onehot_level*, with its metrics depicted in Table 4.4. It was chosen, because of its relatively high test accuracy of 76.5%, combined with a high MCC value of 0.643.

Correctly classified elbows

In this section, six images of correctly classified elbows are included. From these six correctly classified images, one can see that in half of them, the model "looked" outside of the elbow to arrive at its classifications. This is in other words outside the region of interest (ROI). The correctly classified images where the model arrived at its conclusions and was "looking" at the ROI, are given in figure 4.11

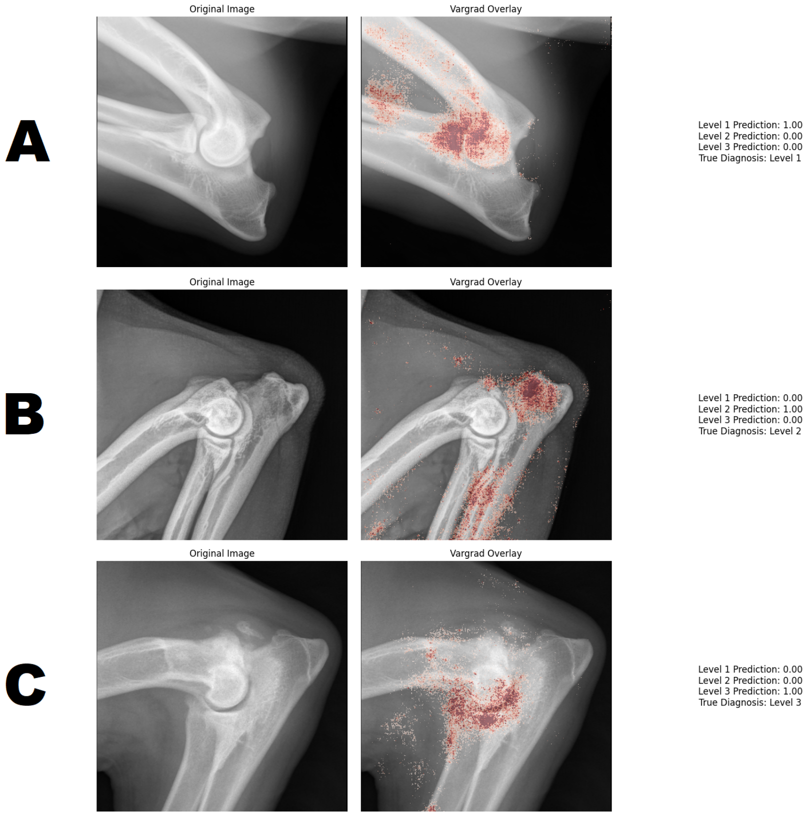


Figure 4.11: X-ray images of elbows with different abnormality levels correctly classified by the *b3_0005_categorical_onehot_level* model. Each panel (A-C) corresponds to true diagnostic levels 1 through 3, respectively. The original images are shown alongside their VarGrad overlays, which highlight in red the regions most influential in the model's predictions. The predicted class probabilities are shown on the right, above the true diagnosis.

Note that in figure 4.11, the images A through C are all correctly classified with a confidence of 100% and that the model was looking at the ROI.

In figure 4.12, the model correctly classified which diagnosis level the elbows belonged to, but looked outside the ROI. It is important to note in figure 4.12 that the model predicted the diagnoses correctly, but was looking outside the ROI. However, the predictions were not as confident as for the images in figure 4.11. The confidence for the classification of image A in figure 4.12 was 91%, 96% for image B and 67% for image C.

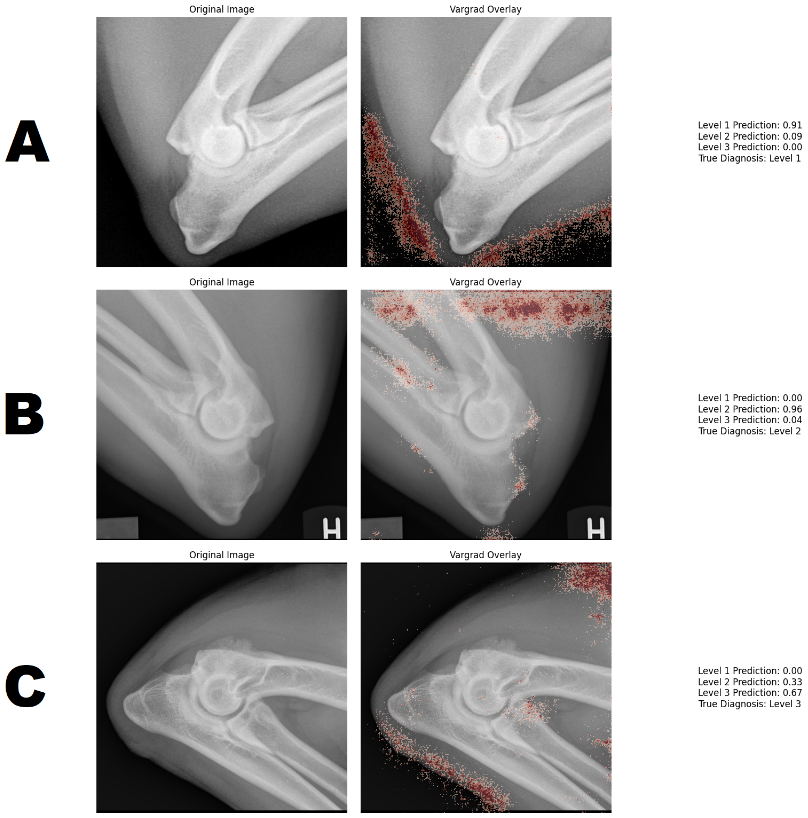


Figure 4.12: X-ray images of elbows with different abnormality levels correctly classified by the *b3_0005_categorical_onehot_level* model. Each panel (A-C) corresponds to true diagnostic levels 1 through 3, respectively. The original images are shown alongside their VarGrad overlays, which highlight in red the regions most influential in the model's predictions. The predicted class probabilities are shown on the right, above the true diagnosis.

Misclassified elbows

Below in figure 4.13, there are three examples of elbows which were misclassified. The model looked at the ROI, but arrived at the wrong classifications. Also note that image A and C in figure 4.13 were both misclassified as level 2, while image B was misclassified as level 1, even though its true class was level 2. This is consistent with what the violin plot in figure 4.10 depicts. Furthermore, it is important to mention that the confidence of the predictions were 99%, 100% and 60% for image A through C, respectively. For instances where the model misclassified and looked outside the ROI, see Appendix B.

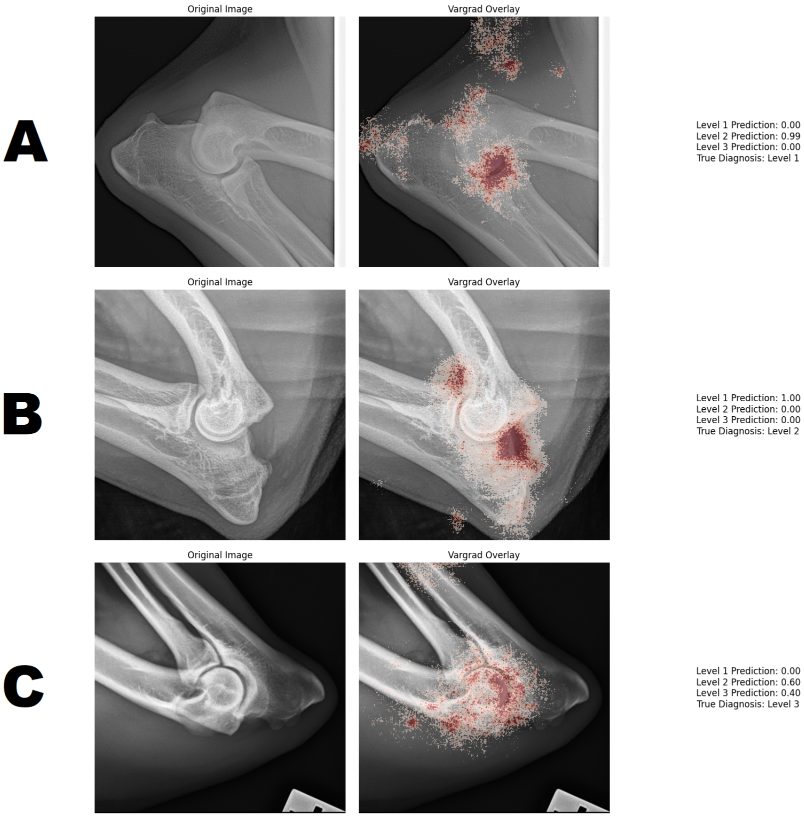


Figure 4.13: X-ray images of elbows with different abnormality levels misclassified by the *b3_0005_categorical_onehot_level* model. Each panel (A-C) corresponds to true diagnostic levels 1 through 3, respectively. The original images are shown alongside their VarGrad overlays, which highlight in red the regions most influential in the model's predictions. The predicted class probabilities are shown on the right, above the true diagnosis.

Chapter 5

Discussion

5.1 Model performance and behavior

This master's thesis advances previous work done by training and assessing various EfficientNet models on canine elbow dysplasia x-ray images. Building on Steiro's results, which garnered a binary accuracy of over 95% and an MCC score of 0.91 for prediction of normal and abnormal elbows [2], the current project has realized a modest increase in multi-class classification accuracy. The project started out with the hypothesis that the loss function used in the training was not optimal, therefore the first objective was to test different loss functions and compare the results. This was, however, not entirely true, because all the loss functions tested in this thesis perform approximately the same, except for the weighted kappa loss function, which performed significantly worse than the rest [32]. A possible reason for the poor performance, could be that it was not implemented optimally in terms of parameters, but this was not explored further.

Presented in table 4.4 is a comparison of the most accurate models: the regression model (*b2_0005_linear_level*) and the highest-performing three-class classification model (*b3_0005_categorical_onehot_level*), which exhibit similar levels of test accuracy. The regression model employed MSE as its loss function, which quantifies the average squared discrepancies between predicted values and actual class labels [26]. This approach generally penalizes large deviations uniformly but does not inherently discriminate based on the classification confidence. In addition, regression models can generalize better if the dataset does not have a complex decision boundary. Conversely, the classification model utilizes categorical cross-entropy, which specifically targets the probability distribution of class labels, maximizing the log probability of the correct class and imposing severe penalties for confident but incorrect predictions [31]. This difference in loss function mechanics subtly influences their performance, with each method's effectiveness potentially hinging on how well it aligns with the nuances of the data and the generalizability of the model. The current hypothesis is therefore that the data was better suited for classification models using categorical cross-entropy as the loss function.

A derived three-class dataset was created in February 2024, which excluded the normal class due to the first few EfficientNet model's high accuracy in correctly classifying it. The process further was, therefore, to try pre-processing the images in the three-class dataset differently. The two methods that were tried out were using three image channels and binarizing the data where two out of the three classes were merged. As seen in figure 4.4, this pre-processing did not increase the performance when compared to the highest overall performing three-class model. It is true that *b1_0005_encode_level* had a higher test accuracy and test MCC-score than *b3_0005_categorical_onehot_level*, but there are two reasons why it is not "better"; When the most accurate encode model was compared to the other encode models, the difference in test accuracy and test MCC-score was large, while being the least complex of them. This was while the validation accuracy was lower than the more complex variants. There is therefore a suspicion that this model was just "lucky". The second reason reason is that a model which can classify each of the three classes of abnormal elbows without merging two of them, has more value for the veterinarians, than a binary model which combines two of the abnormal classes where each class is already comprised of several specific diagnoses each [5]. However, one could argue that either the B1 or B3 variants of the encode models given in table 4.3 could be explored further with explainability analysis, because of their high test accuracy for level 1-2 vs level 3. This could prove to be of value for the veterinarians for efficiently classifying the most severe cases of elbow dysplasia as support for the human reader [5].

Moreover, comparing the highest performing models across different class configurations underscored the importance of aligning model complexity with the dataset's volume and structure. Notably, the four-class model, *b4_0005_categorical_onehot*, which had a complexity level of B4 and used a dataset of 7229 images, demonstrated that larger datasets could support higher complexities, leading to improved performance. Conversely, the three-class model, *b3_0005_categorical_onehot_level*, with a complexity level of B3 and a dataset size of 3030 images, showed a different balance between complexity and dataset size. This finding does not imply that higher complexity results in better performance, but rather that higher complexities tended to perform better when supported by larger datasets.

Moving on, the final step in the process was therefore to choose a model to implement explainability on. The chosen model ended up being the highest overall performing three-class model (*b3_0005_categorical_onehot_level*). However, as seen from several of the results, most notably in figure 4.12, the model cannot be used reliably. This is because even though the model often arrived at the correct predictions, it tended to look outside the intended ROI (the elbow joint). This does not make sense from a radiologist's perspective, because it appears that the model is "guessing".

The tendency of the model to misdirect its focus outside of the ROI could stem from an inadequate volume of representative data for abnormal elbows, limiting the model's ability to learn nuanced distinctions among the classes. This challenge is akin to the difficulty veterinary radiologists face in classifying elbow dysplasia, highlighting the intricate nature of the task that requires not only identifying relevant features but also understanding a complex anatomical context [16].

However, considering the balanced distribution of the training and validation sets, it was less likely that class imbalance was causing the model to incorrectly focus on areas outside of the ROI. Instead, the issue may lie in the intrinsic complexity of the condition and its manifestation in x-rays [1]. The variations in severity and the presence of conditions like arthrosis and sclerosis within the same severity levels could contribute to the model's confusion, reflecting the multifaceted nature of elbow dysplasia [1].

The model's interpretative process could be further obscured by the overlap of features between different severity levels, especially when more common pathologies like arthrosis appear across the spectrum [1]. This suggests that while the model had access to a wide range of examples within each category, the distinguishing features were not as pronounced or as consistently represented as needed for clear differentiation.

A plausible explanation for the model's tendency to focus outside the ROI could be the lack of clear, distinctive features within the ROI itself. If the model fails to identify strong diagnostic cues within the expected anatomical areas, it may "search" for alternative hints in surrounding regions. This behavior indicates that the model is essentially compensating for ambiguous or insufficient data by looking for patterns that might not be directly related to the actual condition but appear to offer clues about the diagnosis. Such a strategy, while creative, underscores the model's limitations in dealing with complex medical imaging tasks.

5.2 Comparison with other papers

In this section, the term the *four-class model* refers to the highest overall performing model among the four-class configurations, and similarly, the *three-class model* refers to the highest overall performing model in the three-class category. These designations are used to enhance readability and preciseness in comparing the models with those discussed in other studies.

The exploration of multi-class classification within the domain of veterinary radiology, particularly for canine elbow dysplasia, is relatively uncharted. This contrasts with more common studies focused on single pathology detection in dogs, such as the investigation into stifle joint diseases [42]. Shim et al.'s work developed deep learning models to diagnose various stifle joint diseases in dogs, achieving diagnostic accuracies that rivaled those of trained veterinarians [42]. The models assessed conditions like patellar deviation and joint effusion, with the advanced object detection capabilities highlighting the potential of AI in veterinary diagnostics.

In their investigation, Shim et al. utilized advanced hardware and sophisticated deep learning architectures to enhance diagnostic capabilities in veterinary medicine. Specifically, the study employed a combination of Faster Region-Based Convolutional Neural Network (Faster R-CNN) for object detection and a 152-layer Residual Network (ResNet) for disease classification [42]. These models were executed on a high-performance platform featuring a NVIDIA GeForce RTX 2070 GPU and an Intel i5-9600KF processor operating at 3.7 GHz. The dataset comprised 2,382 radiographic images of canine stifle joints, processed with a batch size of 1 for Faster R-CNN and 20 for ResNet. The images were divided into training and validation sets with an 80:20 split ratio, and the models were trained for 20,000 and 2,000 epochs respectively [42].

Unlike the binary classification approach utilized by Shim et al, which primarily distinguished between normal and abnormal conditions without further differentiation [42], this thesis introduces nuanced multi-class models. Here, the complexities increased as the models not only identified the presence of an abnormality but also categorized the severity level of elbow dysplasia into multiple classes.

Comparatively, the use of architectures like Faster R-CNN and ResNet-152, as discussed in Shim et al.'s study [42], would potentially result in lower accuracies in the context of this thesis. Faster R-CNN, while robust in object detection scenarios, inherently operates at a slower inference speed than RetinaNet, due to its two-stage detection process [43]. It first generates proposals for objects, then classifies them. This method, while effective for precise object localization, tends to be slower in inference due to the sequential nature of the proposal and classification stages [43].

In contrast, RetinaNet operates using a single-stage detection mechanism, which directly predicts object classes and bounding boxes without the intermediate step of proposal generation [11]. This approach not only speeds up the detection process but also simplifies the model pipeline, making it more suitable for real-time applications [43].

Moreover, EfficientNet architectures demonstrate superior performance with significantly fewer parameters compared to ResNet-152, as evidenced by Tan and Le [12]. This can be viewed in figure 2.18, where one of the least complex models from the EfficientNet family (EfficientNet-B1) achieved a Top 1 accuracy of 79.1% with 7.8 million parameters, while ResNet-152 got 77.8% accuracy with 60 million parameters on ImageNet. These insights highlight the advantages of leveraging EfficientNet, which not only achieved higher accuracies but also enhanced computational efficiency compared to ResNet-152 [12].

For Shim et al.'s study reported specific performance metrics for each diagnosed condition seen in table 5.1. Note that this is four different models using binary classifications, not four distinct classes in one model:
However, a direct comparison of accuracy, sensitivity, and specificity between the four-class model in this thesis and those of Shim et al. is nuanced by two factors:

- **Class Distribution and Real-World Application**: The high prevalence of normal elbows in this thesis' test set mirrored real-world conditions where most dogs assessed do not suffer from elbow dysplasia. While this enhances the

Table 5.1: Comparison of deep learning-based classification models in Shim et al.'s study [42].

Condition	**Accuracy (%)**	**Sensitivity (%)**	**Specificity (%)**
Patellar Deviation	93.18	89.7	96.51
Drawer Sign	81.25	79.41	82.61
Osteophyte Formation	86.49	81.36	89.89
Joint Effusion	85.89	87.50	84.44

model's utility in general screenings, it may also inflate performance metrics compared to Shim et al.'s study, where each condition was balanced between normal and abnormal states. The four-class model in this thesis, demonstrated a test accuracy of 95.8% found in table 4.4, macro sensitivity of 0.7845, and macro specificity of 0.8706 given in table 4.5, suggesting robust performance in identifying normal conditions and differentiating between severity levels of abnormal conditions. Note that the macro sensitivity and specificity for this thesis' four-class model were overall slightly worse than the same metrics presented by shim et al in table 5.1.

- **Complexity of Task**: The multi-class nature of the four-class model introduced a higher level of complexity, as it must not only recognize the presence of an abnormal condition but also discern between different severity levels. This complexity was a significant escalation over binary classification, which may contribute to differences in model performance and applicability.

Given these considerations, the four-class model in this thesis demonstrated significant potential for broad clinical application by efficiently screening out normal cases. Furthermore, the three-class model introduced a different layer of complexity and specialization by focusing exclusively on varying levels of abnormal elbows, excluding the normal class that significantly influenced the performance metrics of the four-class model. This model, therefore, offers insights into the more challenging aspect of diagnosing different severities of elbow dysplasia without the buffer of high accuracy derived from predominantly normal cases.

The three-class model achieved a test accuracy of 76.4% seen in table 4.4, with a macro sensitivity of 76.15% and a macro specificity of 87.00%, as shown in table 4.5. When compared to the specific pathologies in Shim et al.'s study given in table 5.1, the model demonstrated comparable specificity but generally lower sensitivity and accuracy. This is indicative of the increased diagnostic challenge when differentiating between closely related abnormal conditions.

In Shim et al.'s study, even with a binary classification setup, certain conditions like Drawer Sign presented challenges, achieving an accuracy of 81.25% with a sensitivity of 79.41% and specificity of 82.61% seen in table 5.1. This aligns with the inherent difficulty observed in the three-class model where distinguishing between different levels of elbow dysplasia presented a significant challenge.

The three-class model also struggled in particular to distinguish elbows of abnormal level 2 from the other classes. As presented earlier in section 4.1.2, the confusion matrix in figure 4.9, shows a total misclassification rate involving level 2 at 30%. This is further illustrated in the accompanied violin plot in figure 4.10. When looking at these figures, one can draw an understanding to why several of the VarGrad images depicted the most influential parts of the x-ray image as outside the ROI. Often the abnormalities were either subtle, or there were several abnormalities in the elbow at the same time, which could potentially have confused the model. This was in addition to the model already struggling with cases of abnormal level 2.

These comparisons underscore the nuanced capabilities and limitations of multi-class classification systems in veterinary imaging. They also highlight the potential need for enhanced data collection strategies that might include more granular differentiation of disease stages or improved imaging techniques to better capture distinctive features necessary for higher classification accuracy.

Steiro's master's thesis [2] explored similar themes in veterinary imaging, focusing mainly on binary classification, but included also multi-class classification systems for diagnosing canine elbow dysplasia. Steiro's approach utilized several models, but there were two which are directly comparable to this present thesis: a binary classification model distinguishing between normal and abnormal elbows, and a three-class model classifying different severity levels of elbow dysplasia (Nivå 1 vs. 2 vs. 3).
Steiro's binary model achieved good results shown in table 5.2, with an accuracy of 95.6%, an MCC of 0.912, and an AUC of 0.988. In this thesis, the four-class model, while incorporating more complexity by differentiating among four classes, achieved a similar high level of accuracy in distinguishing normal elbows, reflected in its test accuracy of 95.8%, illustrated in table 4.4. This demonstrates a consistency in high performance when it comes to identifying normal conditions, akin to the binary classification in Steiro's study.

Table 5.2: Performance metrics of Steiro's [2] classification models for canine elbow dysplasia.

Model Type	**Accuracy (%)**	**MCC**	**AUC**
Binary Normal/Abnormal	95.6	0.912	0.988
Three-Class (Nivå 1 vs. 2 vs. 3)	66.8	0.502	0.845

For the three-class models, where the task involved discerning among multiple degrees of elbow dysplasia severity:

- Steiro's three-class model reported an accuracy of 66.8%, an MCC of 0.502, and a high AUC of 0.845.

- In comparison, this thesis' three-class model showed a significant improvement, with a test accuracy of 76.4%, test MCC of 0.643, and AUC of 0.885, shown in table 4.4.

The increased accuracy and MCC in the three-class model of this thesis highlight improvements in model sensitivity and the ability to distinguish between the three levels of elbow dysplasia more effectively than Steiro's model. The higher AUC value also indicates a better overall performance in the ROC curve analysis, suggesting this thesis' model's improved reliability in diagnosing varying degrees of elbow dysplasia.

These comparisons with Steiro's work underline the advancements in model performance and diagnostic capabilities over the past year. The current approach not only maintained high standards in binary classification tasks, but also showed significant improvements in the accuracy and reliability of multi-class classifications, highlighting the evolving potential of AI in veterinary diagnostics.

A reason for this improvement could be that the dataset in this thesis was larger than when Steiro worked on her thesis. For instance, the loss function Steiro used on her best three-class model was the same as in this thesis (categorical cross-entropy), the optimal learning rate was also the same (0.0005), consequently, also the model complexity (B3) [2]. The only difference was the size of the dataset.

Another study relevant to this thesis, is the paper by Meetschen et al. [10], which explores the use of Boneview, a diagnostic software by Gleamer based on Facebook AI Research's Detectron 2 framework [44]. Boneview is an AI-enabled fracture detection tool, trained with a dataset of 60 170 trauma radiographs collected from 22 different institutions [45]. Detectron 2, the technology behind Boneview, is an open-source object detection and segmentation platform built on PyTorch. It is notable for its extensive library of pre-trained models and its design that emphasizes modularity and customization, allowing researchers to experiment with various model architectures and functionalities [44].

Boneview is currently in use at Bærum hospital [46]. In Meetschen et al., there was included 200 radiographic images of various regions of the human body [10]. Here, half of the images displayed at least one fracture. Those images were analyzed to assess AI's performance in detecting fractures. Unlike the approach in this master's thesis, the Meetschen et al. study conducted binary classification, determining the presence or absence of fractures. Their AI model achieved a standalone sensitivity of 93% and a specificity of 77% [10].

In contrast, this thesis utilized a multi-class classification approach, which is inherently more complex due to the inclusion of multiple categories for elbow dysplasia severity. The results from the most accurate models, as shown in table 4.5, indicated that multi-class models can achieve comparable, if not superior, performance under certain conditions. Specifically, the four-class model exhibited a macro sensitivity of 0.7845 and a macro specificity of 0.8706, while the micro sensitivity reaches up to 0.9612 and micro specificity up to 0.8151. This demonstrates robust performance across multiple classes, which is particularly notable given the additional complexity involved in distinguishing between multiple levels of severity compared to a binary fracture/non-fracture classification.

Moreover, the three-class model, although slightly less sensitive than the four-class model with a macro sensitivity of 0.7615, showed a strong micro specificity of 0.8709. A plausible reason for this is that classifying abnormal elbows is inherently more difficult than only dividing between normal and abnormal elbows. These findings suggest that the multi-class models are relatively adept at providing nuanced insights into various stages of elbow dysplasia. This ability to differentiate between different levels of disease severity demonstrates the potential of multi-class classification systems in enhancing diagnostic accuracy in veterinary medicine.

However, it is crucial to address the explainability of the models, particularly the three-class model used in this thesis. Despite its high specificity, the reliability of this model is under scrutiny due to its occasional focus outside the intended ROI. This behavior suggests that while the model predicted correctly, it may not always base its decisions on anatomically relevant features, implying a reliance on indirect or possibly spurious patterns not directly associated with the underlying pathology.

5.3 Limitations

This thesis has made advancements in the application of deep learning for diagnosing canine elbow dysplasia. However, several limitations have been identified that could influence the overall effectiveness and applicability of the developed models. These limitations are largely centered around the nature of the dataset and the diagnostic criteria used by veterinarians.

A major limitation encountered in this thesis stems from the imbalanced distribution of the dataset. Of the 7,229 x-ray images analyzed, 4,199 images were of normal elbows, leading to a significant imbalance where normal elbows are overrepresented. More critically, there is an even greater scarcity of images representing specific pathologies such as MCD, UAP, and OCD, which are crucial for training the models to recognize and classify less common but clinically significant conditions. This skewed distribution poses several challenges:

- **Model Bias:** There is an inherent risk of the four-class model developing a bias towards diagnosing elbows as normal, potentially compromising the sensitivity required to detect and classify less prevalent diseases accurately.
- **Generalization Capability:** The overrepresentation of normal elbows and underrepresentation of critical abnormal conditions might limit the models' ability to generalize to a broader, more clinically varied population.

Furthermore, the methodology employed by veterinarians in diagnosing elbow dysplasia involves classifying the condition based on the most severe abnormality present [4]. For example, if an elbow displays signs of both mild sclerosis and more pronounced MCD, the diagnosis will focus on MCD, classifying it as a higher severity grade. This practice can introduce confusion for the model, particularly when it is trained to recognize and differentiate between various stages and types of diseases based on their imaging characteristics.

Veterinarians also typically evaluate more than one elbow image and often compare both elbows of a patient to identify subtle differences that might not be evident when viewing images in isolation [4]. This comparative approach helps in enhancing the accuracy of diagnosing subtle or early-stage diseases, a methodological nuance that the current models do not replicate. Relying on single-image evaluations without the context of comparative anatomy may reduce the model's diagnostic effectiveness, particularly for subtle and complex conditions.

5.4 Further work

Even though the work done in this thesis shows potential, there are several aspects which could be improved upon and explored further. This is to ensure that models can be reliable and of clinical importance.

A start could be to develop CNNs which utilize x-ray images of both the right and left elbows of dogs. This dual image approach is akin to how veterinarians diagnose ED, which could also make it easier to spot subtle differences [4]. Consequently, implementing an algorithm that automatically selects the most severe level of ED when the prediction probabilities across different classes are closely matched, could help resolve the model's confusion in cases with multiple coexisting abnormalities.

Furthermore, one could experiment with vision transformers [47] with advanced augmentation techniques such as RandAugment [48] or AutoAugment [49] to see if there is an improvement in performance for classifying ED. This could potentially improve the performance, however as stated by Steiner et al. [50], this requires an extensive amount of data, greater than what was available during the course of this thesis. Robust data collection and model training strategies is therefore important for this approach. An interesting proposition to explore, could be the pre-training of vision transformers on the extensive dataset of the original, unlabeled x-ray images before fine-tuning them with labeled data [51]. This methodology might lead to advancements in diagnostic accuracy and the robustness of the model.

Chapter 6

Conclusion

This thesis advances the use of deep learning for diagnosing dog elbow dysplasia by employing EfficientNet models of varying complexities and exploring multiple loss functions. Building on Steiro's master thesis, which achieved high binary accuracy and MCC [2], this thesis extended into multi-class classification and observed a modest improvement in accuracy, likely due to the expanded dataset which provided a more robust foundation for training the CNNs.

Although most tested loss functions performed comparably, the weighted kappa loss function underperformed, potentially due to suboptimal parameter implementation. Furthermore, a key focus of the thesis was the experimentation with different image pre-processing methods in a derived three-class dataset that excluded the normal elbow class, because of previous high classification accuracy. Techniques such as using three image channels where two of the channels augmented the original image, and binarizing the data, where two of the three classes were merged, were tested. However, these methods did not improve performance over the highest overall performing three-class model.

Notably, the binarization approach aimed to simplify the model's task by reducing classification complexity. While a multi-class model provides greater diagnostic value, binarization showed potential as supplementary support for classifying the most severe cases of elbow dysplasia. This aspect, though not further explored in this thesis, could be valuable in clinical settings where rapid identification of severe cases is of high interest.

The highest performing three-class model underwent explainability analysis using VarGrad to better understand the model's predictive behaviors. Despite its higher performance, this model demonstrated limitations in reliability. The model frequently focused on areas outside the intended region of interest and misclassified cases of level 2 elbow dysplasia 30% of the time. This misdirection suggests that the model might be "guessing" at times, highlighting the need for further refinement in the model's interpretative capabilities to enhance its clinical applicability and reliability.

Bibliography

[1] Spencer A. Johnston and Karen M. Tobias. *Veterinary Surgery: Small Animal.* 2nd ed. Vol. 1. St. Louis, Missouri: Elsevier, 2018. ISBN: 0323320651.

[2] Sunniva Elisabeth Daae Steiro. "Automatisk deteksjon av abnormaliteter i hundealbuer". MA thesis. Norwegian University of Life Sciences, 2023.

[3] Norsk Kennel Klub. *Regler for praktisering av innførte registreringsrestriksjoner for HD og AD.* `https://www.nkk.no/for-veterinaerer/skjelettlidelser/ad-albueleddsdysplasi/`. Accessed: April 24th, 2024. 2021.

[4] Hege Kippenes Skogmo. *Oral Communication.* Jan. 2024.

[5] Hege Kippenes Skogmo. *Personal Communication.* May 2024.

[6] A. Oberbauer, G. Keller, and T. Famula. "Long-term genetic selection reduced prevalence of hip and elbow dysplasia in 60 dog breeds". In: *PLoS ONE* 12 (2017). DOI: `10.1371/journal.pone.0172918`.

[7] Yue Zhou et al. "Multi-task learning for segmentation and classification of tumors in 3D automated breast ultrasound images". In: *Medical image analysis* 70 (2020), p. 101918. DOI: `10.1016/j.media.2020.101918`.

[8] Chaouki Boufenar et al. "Computer-aided diagnosis of Canine Hip Dysplasia using deep learning approach in a novel X-ray image dataset". In: *Computer Methods in Biomechanics and Biomedical Engineering: Imaging & Visualization* 11.7 (2024), p. 2274947. DOI: `https://doi.org/10.1080/21681163.2023.2274947`.

[9] Erin Hennessey et al. "Artificial intelligence in veterinary diagnostic imaging: A literature review". In: *Veterinary Radiology and Ultrasound* 63.S1 (Dec. 2022), pp. 851–870. DOI: `https://doi.org/10.1111/vru.13163`.

[10] Mathias Meetschen et al. "AI-Assisted X-ray Fracture Detection in Residency Training: Evaluation in Pediatric and Adult Trauma Patients". In: *Diagnostics* 14.6 (2024), p. 596. DOI: `10.3390/diagnostics14060596`. URL: `https://www.mdpi.com/2075-4418/14/6/596`.

[11] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision.* 2017, pp. 2980–2988.

[12] Mingxing Tan and Quoc Le. "Efficientnet: Rethinking model scaling for convolutional neural networks". In: *International conference on machine learning.* PMLR. 2019, pp. 6105–6114.

[13] Brian H Brown et al. *Medical physics and biomedical engineering.* CRC Press, 2001.

[14] Maggie A Flower. *Webb's physics of medical imaging.* CRC press, 2012.

[15] B Tellhelm. "Grading primary ED-lesions and elbow osteoarthrosis according to the IEWG protocol". In: *Proceeding of the Annual Meeting of the Executive Committee of the IEWG.* Ed. by B Tellheim. Amsterdam, Netherlands, 2011.

[16] Aldo Vezzoni and Kevin Benjamino. "Canine elbow dysplasia: ununited anconeal process, osteochondritis dissecans, and medial coronoid process disease". In: *Veterinary Clinics: Small Animal Practice* 51.2 (2021), pp. 439–474.

[17] Donald E. Thrall, ed. *Textbook of Veterinary Diagnostic Radiology.* 7th ed. Elsevier, 2018. ISBN: 9780323482479.

[18] Jacob Michelsen. "Canine elbow dysplasia: aetiopathogenesis and current treatment recommendations". In: *The Veterinary Journal* 196.1 (2013), pp. 12–19. DOI: `https://doi.org/10.1016/j.tvjl.2012.11.009`.

[19] Norsk Kennel Klub. *Prosedyrebeskrivelse ved røntgenfotografering for avlesning i Norsk Kennel Klub.* `https://www.nkk.no/for-veterinaerer/skjelettlidelser/ad-albueleddsdysplasi/`. Accessed: April 24th, 2024. 2015.

[20] A. L. Samuel. "Some Studies in Machine Learning Using the Game of Checkers". In: *IBM Journal of Research and Development* 3.3 (1959), pp. 210–229. DOI: `10.1147/rd.33.0210`.

[21] Sebastian Raschka and Vahid Mirjalili. *Python machine learning: Machine learning and deep learning with Python, scikit-learn, and TensorFlow 2.* Packt Publishing Ltd, 2019.

[22] Francois Chollet. *Deep learning with Python.* Simon and Schuster, 2021.

[23] Marco A Wiering and Martijn Van Otterlo. *Reinforcement learning.* Vol. 12. 3. Springer, 2012, p. 729.

[24] Mohammad Hossin and Sulaiman M.N. "A Review on Evaluation Metrics for Data Classification Evaluations". In: *International Journal of Data Mining Knowledge Management Process* 5 (Mar. 2015), pp. 01–11. DOI: `10.5121/ijdkp.2015.5201`.

[25] Davide Chicco and Giuseppe Jurman. "The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation". In: *BMC Genomics* 21.1 (2020), p. 6. DOI: `10.1186/s12864-019-6413-7`. URL: `https://doi.org/10.1186/s12864-019-6413-7`.

[26] Alexei Botchkarev. "Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology". In: *arXiv preprint arXiv:1809.03006* (2018).

[27] Bradley J. Erickson and Felipe Kitamura. "Magician's Corner: 9. Performance Metrics for Machine Learning Models". In: *Radiology: Artificial Intelligence* 3.3 (2021), e200126. DOI: `10.1148/ryai.2021200126`. eprint: `https://doi.org/10.1148/ryai.2021200126`. URL: `https://doi.org/10.1148/ryai.2021200126`.

[28] Eli Stevens, Luca Antiga, and Thomas Viehmann. *Deep learning with PyTorch.* Manning Publications, 2020.

[29] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* `http://www.deeplearningbook.org`. MIT Press, 2016.

[30] Karl Weiss, Taghi M. Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big Data* 3.1 (May 2016). DOI: `https://doi.org/10.1186/s40537-016-0043-6`.

[31] Pieter-Tjerk De Boer et al. "A tutorial on the cross-entropy method". In: *Annals of operations research* 134 (2005), pp. 19–67.

[32] Jordi de La Torre, Domenec Puig, and Aida Valls. "Weighted kappa loss function for multi-class classification of ordinal data in deep learning". In: *Pattern Recognition Letters* 105 (2018), pp. 144–154.

[33] Bao Ngoc Huynh. "Visualization of deep learning in auto-delineation of cancer tumors". MA thesis. Norwegian University of Life Sciences, Ås, 2020.

[34] Bao Ngoc Huynh. *Cubiai.* `https://github.com/huynhngoc/cubiai`. 2024.

[35] *Orion High Performance Computing.* 2024. URL: `https://orion.nmbu.no/en/home`.

[36] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.* Software available from tensorflow.org. 2015. URL: `https://www.tensorflow.org/`.

[37] Sara Hooker et al. "A benchmark for interpretability methods in deep neural networks". In: *Advances in neural information processing systems* 32 (2019).

[38] M. Abdullah-Al-Wadud et al. "A Dynamic Histogram Equalization for Image Contrast Enhancement". In: *IEEE Transactions on Consumer Electronics* 53.2 (2007), pp. 593–600. DOI: `10.1109/TCE.2007.381734`.

[39] Guang Deng. "A Generalized Unsharp Masking Algorithm". In: *IEEE Transactions on Image Processing* 20.5 (2011), pp. 1249–1261. DOI: `10.1109/TIP.2010.2092441`.

[40] Mauricio Reyes et al. "On the interpretability of artificial intelligence in radiology: challenges and opportunities". In: *Radiology: artificial intelligence* 2.3 (2020), e190043.

[41] NMBU REALTEK. *Retningslinjer for bruk av kunstig intelligens ved REALTEK.* Norwegian University of Life Sciences (NMBU), Faculty of Science and Technology. Available online: `https://www.nmbu.no/fakulteter/fakultet-realfag-og-teknologi/kunstig-intelligens-ved-realtek`. 2024. URL: `https://www.nmbu.no/fakulteter/fakultet-realfag-og-teknologi/kunstig-intelligens-ved-realtek`.

[42] Hyesoo Shim et al. "Deep learning-based diagnosis of stifle joint diseases in dogs". In: *Veterinary Radiology & Ultrasound* 64.1 (2022), pp. 113–122.

[43] Wenwen Li et al. "GeoImageNet: a multi-source natural feature benchmark dataset for GeoAI and supervised machine learning". In: *GeoInformatica* 27 (Sept. 2022). DOI: `10.1007/s10707-022-00476-z`.

[44] Yuxin Wu et al. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[45] J. Oppenheimer et al. "A Prospective Approach to Integration of AI Fracture Detection Software in Radiographs into Clinical Workflow". In: *Life (Basel, Switzerland)* 13.1 (2023), p. 223. DOI: 10.3390/life13010223. URL: https://doi.org/10.3390/life13010223.

[46] Kvalnes, P. *KI på norske sykehus: – Det kommer til å bli annerledes enn vi tror*. Accessed: 2024-05-01. 2024. URL: https://www.oslomet.no/forskning/forskningsnyheter/ki-norske-sykehus.

[47] Alexey Dosovitskiy et al. "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale". In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=YicbFdNTTy.

[48] Ekin D. Cubuk et al. *RandAugment: Practical automated data augmentation with a reduced search space*. 2019. arXiv: 1909.13719 [cs.CV].

[49] Ekin D. Cubuk et al. *AutoAugment: Learning Augmentation Policies from Data*. 2019. arXiv: 1805.09501 [cs.CV].

[50] Andreas Steiner et al. "How to train your vit? data, augmentation, and regularization in vision transformers". In: *arXiv preprint arXiv:2106.10270* (2021).

[51] Leo Quentin Bækholt. *Oral Communication*. May 2024.

Appendix A

Confusion matrices for regression models

This appendix presents confusion matrices for the four regression models (B1 - B4) analyzed in this thesis. Each model corresponds to a different level of model complexity. For each model, two matrices are displayed: one for the validation set and one for the test set. These matrices are arranged side-by-side to facilitate direct comparison, with the validation matrix on the left and the test matrix on the right.

The matrices illustrate the accuracy and misclassification rates for each model across different severity levels of elbow dysplasia. This arrangement provides a comprehensive view of how each model's complexity influences its performance, particularly highlighting differences between validation and testing phases.

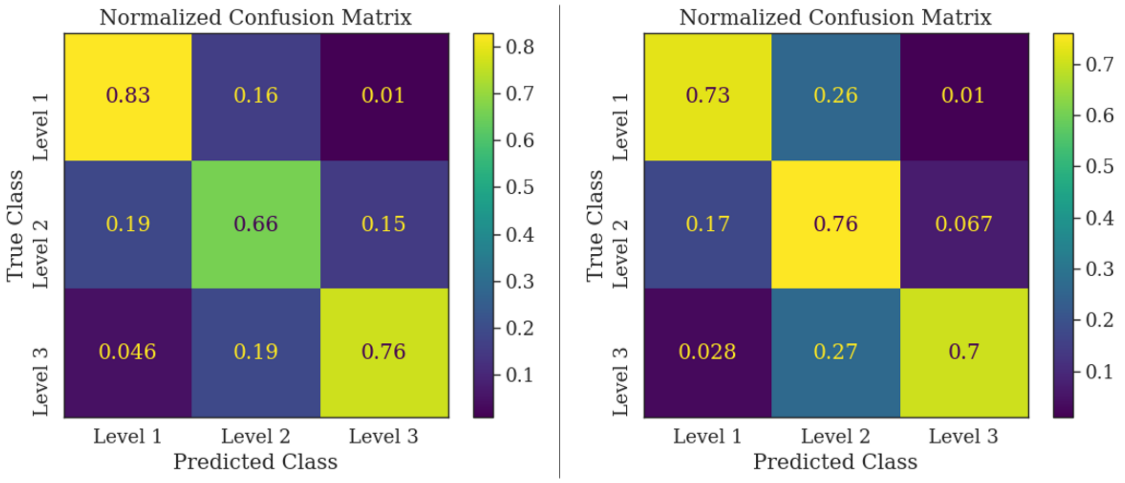


Figure A.1: Complexity B1

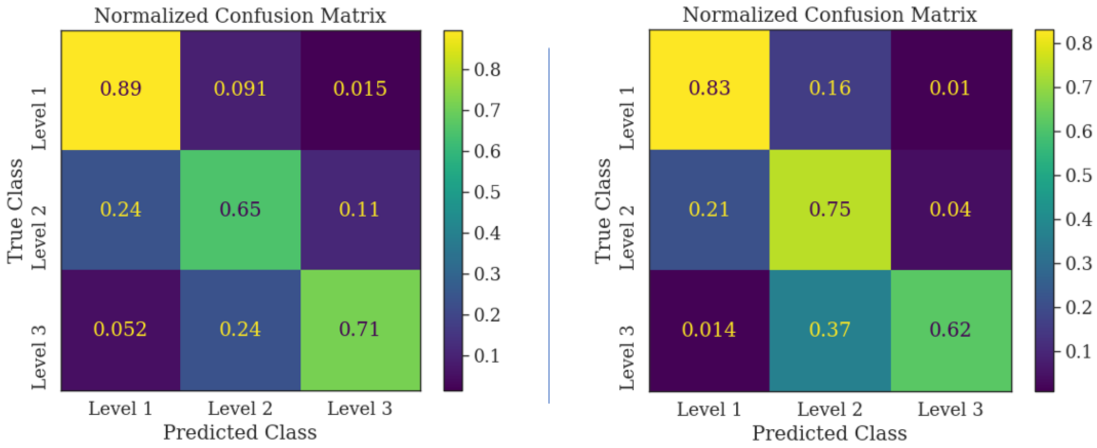


Figure A.2: Complexity B2

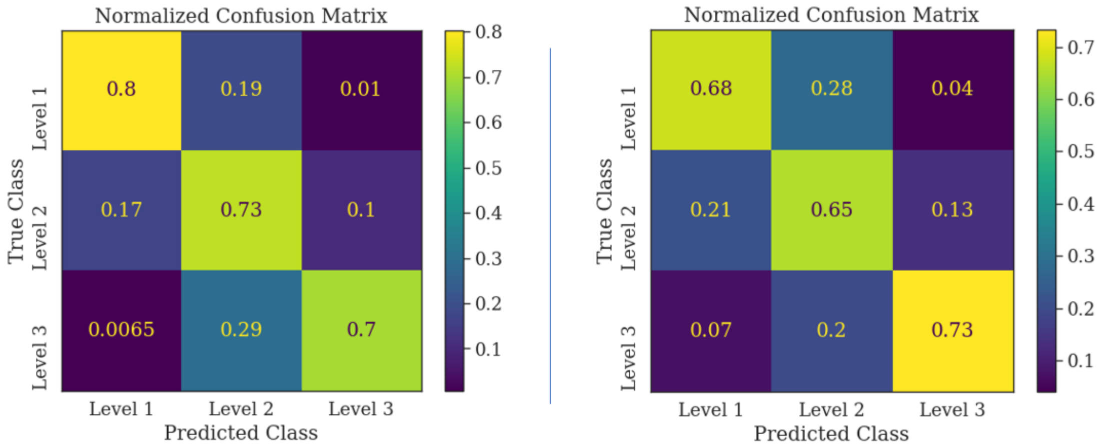


Figure A.3: Complexity B3

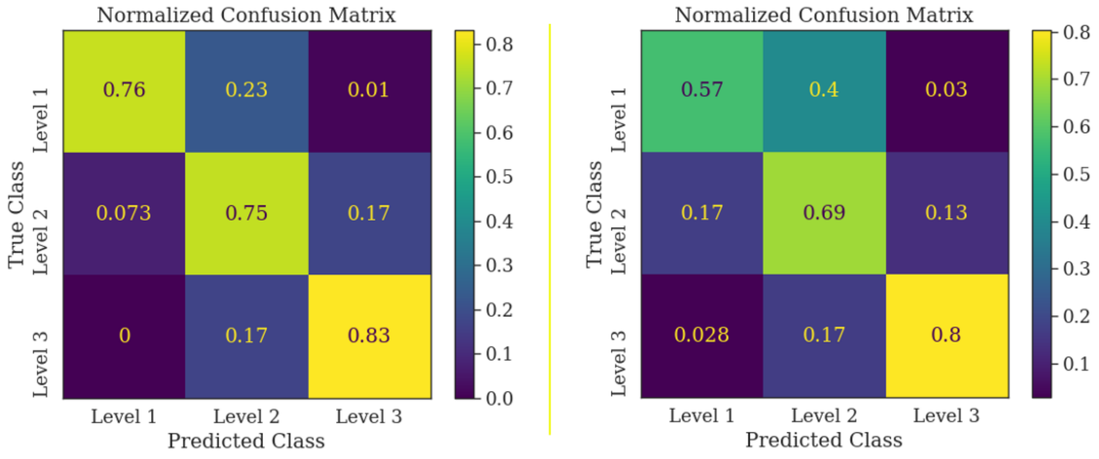


Figure A.4: Complexity B4

Appendix B

Misclassified images with VarGrad overlay (outside the intended region of interest)

This appendix features a collection of misclassified images from the three-class model that had VarGrad implemented. These images are each accompanied by a VarGrad overlay. The images are selected to specifically showcase instances where the model's attention was not correctly focused on the intended region of interest (the elbow joint).

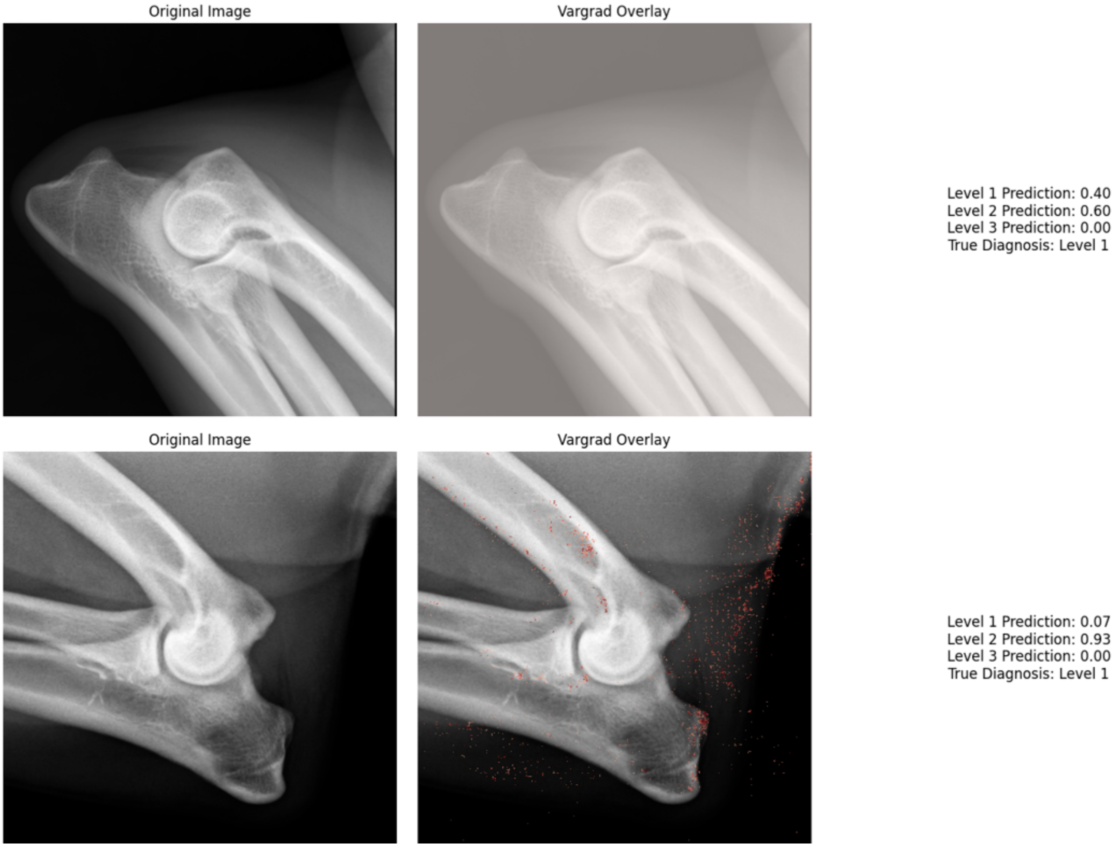
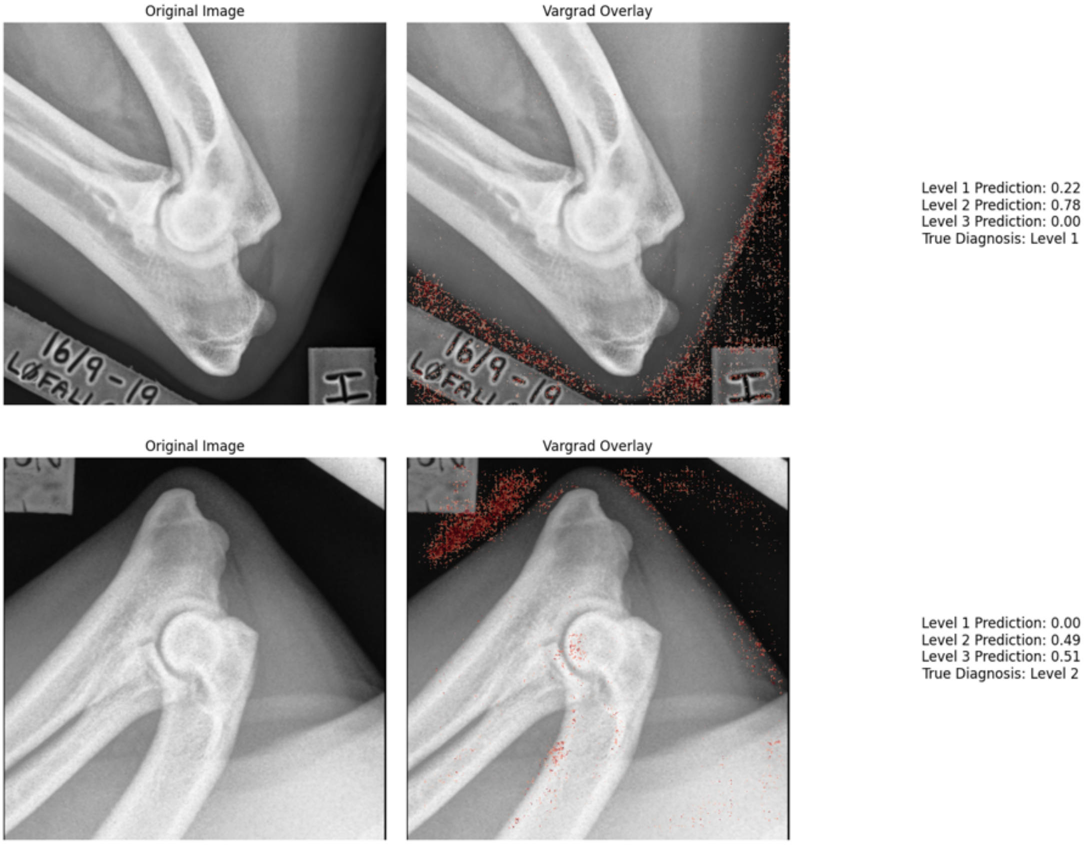


Figure B.1



Level 1 Prediction: 0.22
Level 2 Prediction: 0.78
Level 3 Prediction: 0.00
True Diagnosis: Level 1

Level 1 Prediction: 0.00
Level 2 Prediction: 0.49
Level 3 Prediction: 0.51
True Diagnosis: Level 2

Figure B.2

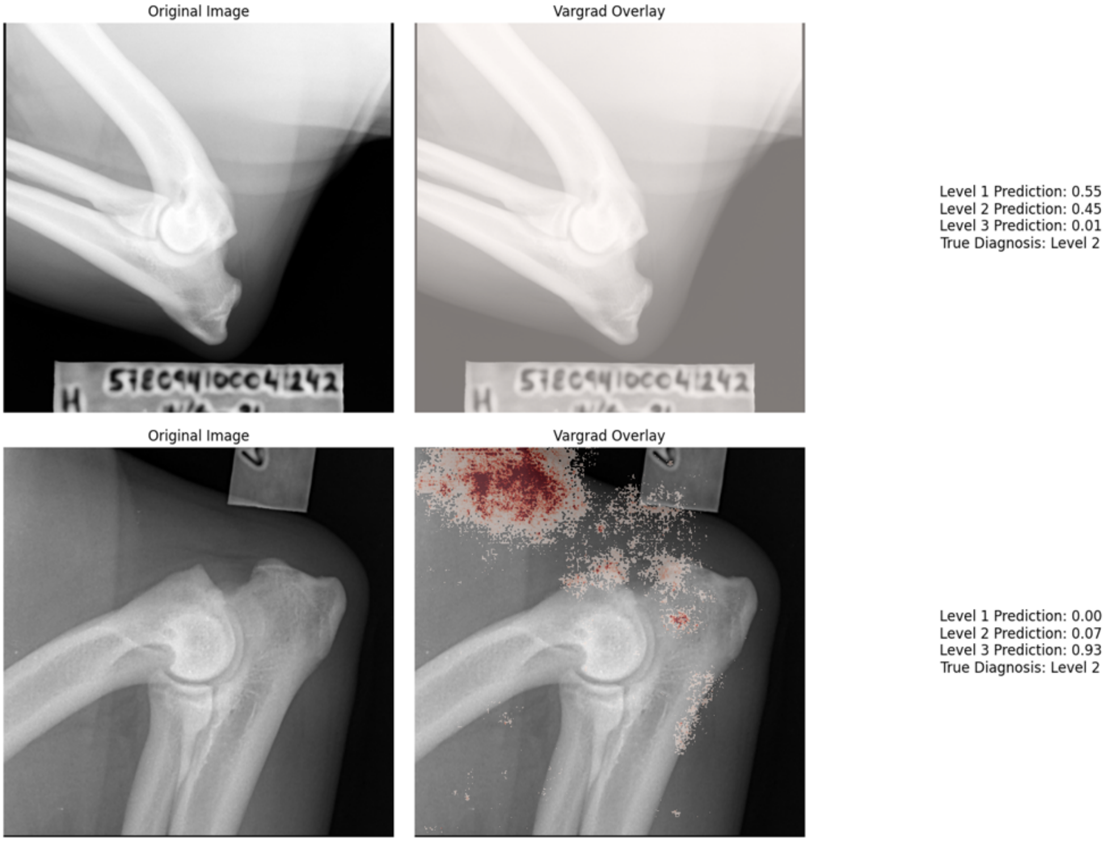

Original Image
Vargrad Overlay
57EC9410004242
H
57EC9410004242
H
Level 1 Prediction: 0.55
Level 2 Prediction: 0.45
Level 3 Prediction: 0.01
True Diagnosis: Level 2
Original Image
Vargrad Overlay
Level 1 Prediction: 0.00
Level 2 Prediction: 0.07
Level 3 Prediction: 0.93
True Diagnosis: Level 2

Figure B.3

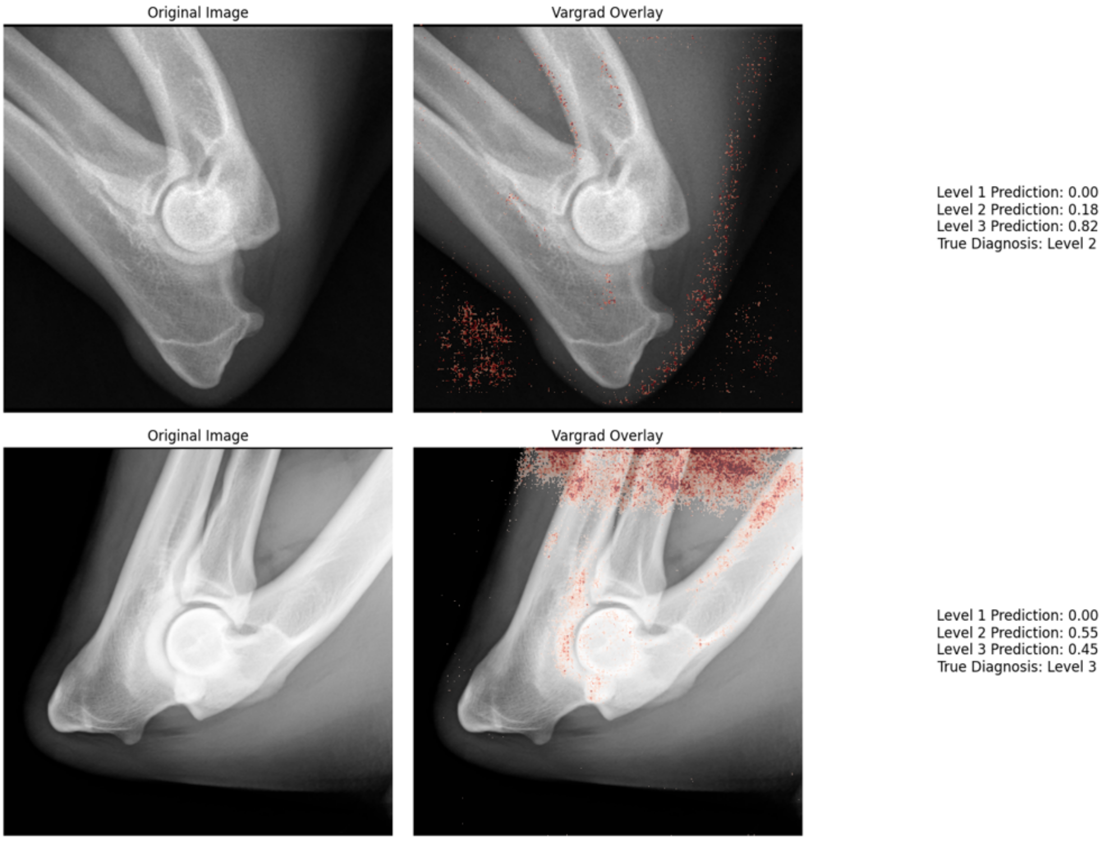

Original Image
Vargrad Overlay
Level 1 Prediction: 0.00
Level 2 Prediction: 0.18
Level 3 Prediction: 0.82
True Diagnosis: Level 2
Original Image
Vargrad Overlay
Level 1 Prediction: 0.00
Level 2 Prediction: 0.55
Level 3 Prediction: 0.45
True Diagnosis: Level 3

Figure B.4

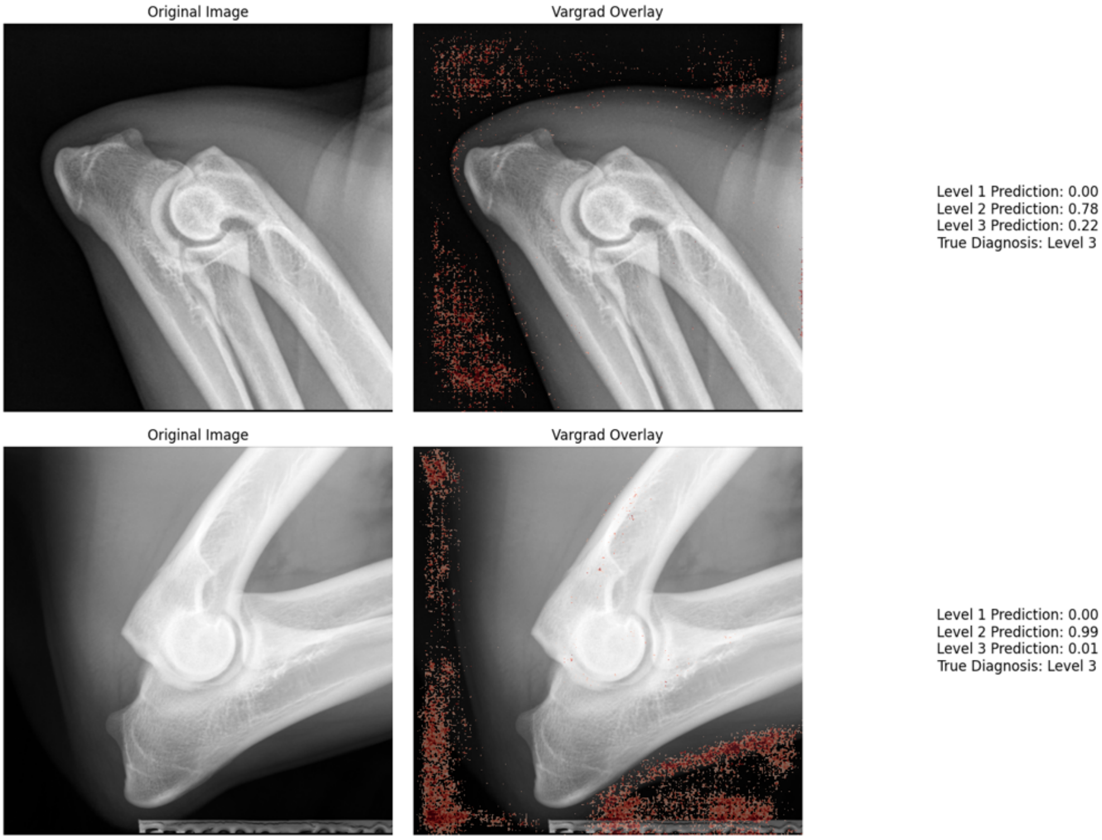

Original Image
Vargrad Overlay
Level 1 Prediction: 0.00
Level 2 Prediction: 0.78
Level 3 Prediction: 0.22
True Diagnosis: Level 3
Original Image
Vargrad Overlay
Level 1 Prediction: 0.00
Level 2 Prediction: 0.99
Level 3 Prediction: 0.01
True Diagnosis: Level 3

Figure B.5

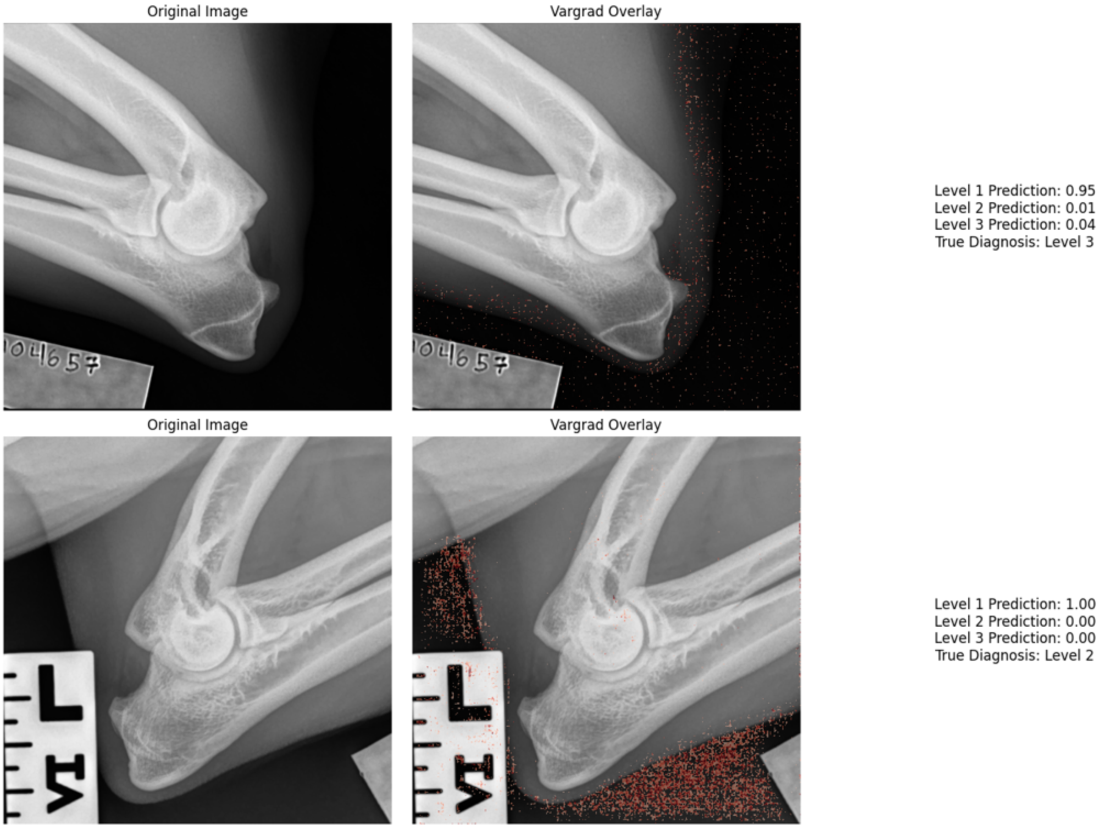

Original Image
Vargrad Overlay
Level 1 Prediction: 0.95
Level 2 Prediction: 0.01
Level 3 Prediction: 0.04
True Diagnosis: Level 3
Original Image
Vargrad Overlay
Level 1 Prediction: 1.00
Level 2 Prediction: 0.00
Level 3 Prediction: 0.00
True Diagnosis: Level 2

Figure B.6

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway