Interpretable Classification of Canine Emotional Behaviour with Computer Vision

Shahmurad Orujov

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the Degree of Master of Science at The University of Glasgow

3 September 2025

Abstract

This project implements, evaluates, and compares interpretable computer vision methods for classifying canine emotional behaviour. The work replicates a prior baseline established by Chávez-Guerrero et al. and adds three additional sets of experiments. First, data augmentation experiments were conducted using MixUp, CutMix, random erasing, mosaic augmentation, and geometric transformations. Second, new model architectures, including EfficientNetV2-M, ConvNeXt-Base and DINOv2 ViT-S/14, were trained with AdamW optimizer, cosine annealing learning rate scheduler and progressive unfreezing. Third, a combined dataset was created by merging three datasets to improve class balance and data diversity. Then, the dataset was used in experiments with six neural network architectures under the unified pipeline. Evaluation for each experiment reports accuracy, macro-F1, weighted-F1, per-class scores, and confusion matrices. Additionally, the interpretability of predictions is assessed by overlaying the EigenCAM heatmaps on the original images, providing insight beyond purely quantitative metrics. On the original dataset, MixUp and CutMix increase accuracy, lifting MobileNetV2 from 0.6937 to 0.7250 and ResNet50 from 0.6687 to 0.6875. On the combined dataset, the strongest results are obtained from modern architectures, with DINOv2 ViT-S/14 achieving 0.8621, ConvNeXt-Base reaching 0.8592, and EfficientNetV2-M attaining 0.8545. Per-class F1 scores show that the models generalise across the classes with modest gaps between most of them. Despite the outstanding accuracy of DINOv2 ViT-S/14, EigenCAM overlays indicate that convolutional networks generally focus on the dog's face and body, whereas the transformer model often spreads attention to the background context. The project delivers an interpretable baseline that ties performance gains to animal-centric reasoning, acknowledging its limitations while providing recommendations and a solid foundation for future work on explainable computer vision in Animal-Computer Interaction.

Keywords: Animal-Computer Interaction; canine emotion classification; interpretable computer vision; EigenCAM; convolutional neural networks (CNNs); vision transformers (ViTs).

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format.

Name: Shahmurad Orujov Signature: Shahmurad Orujov

AI Usage Declaration

In preparing this dissertation, I used Artificial Intelligence (AI) tools in the following limited ways:

- **Language and formatting support:** AI was used to refine grammar, improve clarity and coherence, maintain consistent terminology and style, and resolve general LaTeX configuration and compilation issues. All wording and formatting choices were reviewed and finalised by me.
- **Programming assistance:** AI provided diagnostic guidance and alternative approaches during development in Python and PyTorch. I assessed, adapted, implemented, and validated any suggested changes; the final code reflects my own design decisions and testing, and I take full responsibility for it.
- **Literature scoping:** AI helped to broaden and refine search queries and to consolidate preliminary information for mapping the research landscape. I independently retrieved, read, and critically evaluated all sources; the final selection, interpretation, and citations are entirely my own.

All ideas, arguments, analyses, and conclusions presented in this dissertation are my own. AI was not used for the critical evaluation of sources, for forming my arguments or conclusions, or to produce research outputs. No confidential or personal data was provided to AI systems.

My use of AI complied with the University of Glasgow policy on ethical and transparent AI use in academic work and did not replace my own understanding or critical judgment.

Name: Shahmurad Orujov Signature: Shahmurad Orujov

Acknowledgements

I thank Dr. Ilyena Hirskyj-Douglas for her guidance, patience, and steady support throughout this project. Her mentoring helped me enter the field of Animal-Computer Interaction with a clear, welfare-centred view, and her thoughtful feedback significantly contributed to my understanding of the area.

I am also grateful to the authors of the original study, especially Dr. Humberto Pérez-Espinosa, for kindly sharing the dataset and recommending the DEBiW dataset, which was later included in the experiments. Their prompt and helpful replies to my questions made the replication and the rest of the project realisable in such a short amount of time.

Contents

Chapter 1: Introduction

Animal-Computer Interaction (ACI) considers animals as users and a cornerstone of research, prioritising their welfare, agency, and ethics above all else. Reviews of the ACI field define various types of systems for animals and introduce feedback loops that enable animals to influence the outcomes. Nevertheless, despite its breadth, the field is fragmented, with different methods and evaluation practices, limited channels for direct engagement, and scarce evidence on long-term welfare. In this context, building systems that may positively influence actual decisions about animals requires adequate scoping and clear guidelines.

Emotions in dogs must be inferred from visible signs and behavioural cues, such as posture, head and tail movements, and gaze. Still, such cues can be ambiguous and context-sensitive around class boundaries. On the other hand, physiological measures are informative yet might require invasive design. In many settings, computer vision using images or videos, often implemented with transfer learning, can become the path of least friction. However, recurrent problems with data, labeling consistency, sensitivity to breed and background, and uneven use of interpretability beyond headline accuracy persist and may hinder the fairness of deductions.

This project puts reliability by comparing several widely used CNNs and a Vision Transformer under a unified pipeline, addressing dataset scale issues by combining data from three distinct sources, and using quantitative metrics alongside EigenCAM prediction interpretability to verify whether the models rely on class-specific cues related to dogs rather than taking shortcuts through the background. To summarise the project's goals, four research questions are intended to be investigated and answered in the following chapters.

RQ1 How do representative CNNs and a ViT compare on balanced performance under identical conditions?

RQ2 Do explanations indicate dog-centred attention, or do models rely on background/context shortcuts?

RQ3 Does a combined dataset with a unified four-class scheme improve robustness across sources?

RQ4 Which simple calibration and abstention rules make model outputs more trustworthy for welfare-aligned decisions?

Chapter 2: Literature Review

Before delving into the details of this project's methodology, it is essential to conduct a review of the literature and identify the gaps in animal-computer interactions, particularly in emotion recognition and behaviour analysis in animals.

2.1 Introduction to Animal-Computer Interaction (ACI)

Animal-computer interaction (ACI) is a research field that is devoted to designing and studying interactive technologies for animals as users. The foundations of modern ACI were laid by Mancini's manifesto, which challenged a human-centric approach to animal-related technologies and put animals at the centre of importance while designing interactive systems to make them legitimate stakeholders [1]. To support animal welfare, agency, and develop ethical and animal-centric design frameworks were the main goals established for ACI by the manifesto.

Seven years later, a comprehensive review by Hirskyj-Douglas et al. identified five primary interface types: tangible and physical objects, haptic and wearable technologies, olfactory interfaces, screen-based interfaces, and tracking systems [2]. The review emphasised the importance of designing the feedback loop to allow animals to interact with technologies meaningfully. However, key challenges, including limited understanding of animal agency, difficulties in interpreting animal intentions, and the need for more nuanced design frameworks and long-term studies, remained.

Recent large-scale scoping work by Kleinberger et al. analysed nearly 800 studies, mapping animal technologies across 11 research objectives, 8 technology types, and 6 contexts [3]. Despite the diversity that included work on livestock, wildlife, pets, working animals, primates, and aquatic species, the field remains fragmented, and many systems still offer limited opportunities for direct engagement and meaningful feedback for animals. Furthermore, only about one-third of the studies treated animals as primary beneficiaries or provided feedback mechanisms, indicating persistent ethical challenges regarding agency and consent, and underscoring the ongoing need for more welfare-oriented and empirically validated design approaches within ACI.

2.2 Canine Emotion Recognition within ACI

Now that the foundations and current challenges within ACI have been established, it is necessary to talk about the details of the current project: canine emotion recognition and computer vision.

The central aim of this project is to replicate and attempt to enhance the methodology presented by Chávez-Guerrero et al. for classifying canine emotional behaviour using computer vision [4]. Hence, the following literature review will focus specifically on the intersection of computer vision and emotion recognition within the context of ACI. This focus is motivated by the rapid development of emotional behaviour recognition, for both scientific understanding of animal welfare and human-animal collaboration, as well as the progress achieved in automatically detecting, classifying, and interpreting animal emotions for visual data.

Given that animals, unlike humans, cannot communicate their internal states verbally, it is crucial to understand animal emotions from observable and measurable signals [1]. Theoretical approaches to emotion recognition in animals can be broadly categorised as behavioural, physiological, or multimodal. Behavioural approaches, such as the Dog Facial Action Coding System, highly rely on external indicators, including body posture, facial expression, tail movements, vocalisation, and other observable actions [5]. Physiological approaches, on the other hand, can be invasive and use measurements such as heart rate, hormone level, or brain activity. Multimodal approaches aim to improve reliability by integrating complementary information sources and combining behavioural and physiological data.

Practical implications of accurate emotional recognition range from early identification of stress and anxiety to improved care in contexts such as shelters, working animal environments, and research settings. And since dogs are well known to be common in all of the contexts mentioned above, and even considered to be human companions more often than any other animal, it is of specific interest to accurately recognise their emotional behaviour as well [6].

Various methods, such as observation of body language, tail, posture, ears, facial cues, vocal analysis of barks, growls, whines or physiological signal analysis, are used for emotional recognition, covering the most important theoretical approaches [7]. Although these methods are still imperfect and have flaws like a need for high-quality, well-annotated, large datasets, the particular problems of invasiveness and ambiguity of signals in physiological methods are prompting researchers to look into non-invasive methods using computer vision and machine learning [8].

2.3 Computer Vision Approaches to Dog Emotion Recognition

Recent research on the automation of dogs' emotional behaviour uses images and videos commonly separated into different classes, including anxiety, aggression, happiness, fear, and neutrality [4], [8]. The most common approach is to use convolutional neural networks (CNNs) with transfer learning, utilising the benefits of existing models solving similar classification problems. In contrast with early methods that required manual feature engineering, deep learning allows end-to-end learning from raw data. Popular network architectures featured in recent literature include ResNet, VGG-16, MobileNet, EfficientNet, ViT, and YOLO [4], [9], [10]. Moreover, some research aims not only to get the model to work as desired but also to understand how it works by incorporating such model interpretability techniques as EigenCAM activations [9]. However, despite addressing the invasiveness issues and in addition to the obvious dataset requirements, machine learning approaches introduce new concerns, such as model generalisability, a lack of standardised benchmarks, and sensitivity to breed or context differences.

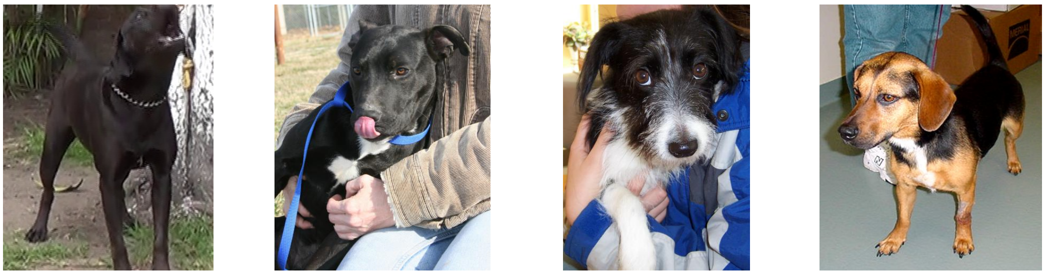
To finalise this chapter, current research gaps need to be identified and clearly stated to have a vivid picture of the project. Firstly, most existing datasets are small, unbalanced and not diverse enough, leading to poor model generalisation and overfitting. Secondly, no standard categories exist, resulting in ambiguous labeling. Thirdly, the focus on model interpretability and explainability is minimal and can be found in only a small portion of publications. Finally, the methodology should be integrated into a unified framework for future consistency, comparability, and reproducibility. Thus, in the next chapter, this project will attempt to replicate and improve upon existing methodologies, employing both data-centric and model-centric approaches, as well as more rigorous evaluation.

Chapter 3: Methodology

3.1 Reproduction of the original work by Chávez-Guerrero et al.

3.1.1 Dataset

The dataset used in this project was obtained by request from the authors of the original paper [4]. It consists of 1,067 images of domestic dogs, originating from annotated video frames, with basic filtering and preprocessing as described in the original work, including standard resizing and normalisation to fit model input requirements. The dataset was later manually labeled and split into four classes: Aggressiveness, Anxiety, Fear, and Neutral. Examples of these images can be seen in Figure 3.1. Additionally, basic data augmentations, such as random horizontal flipping, rotations of up to 20 degrees, and adjustments to brightness and contrast, were applied to the images [11].



(a) Aggressiveness (b) Anxiety (c) Fear (d) Neutral

Figure 3.1: Examples of different emotional states from the dataset corresponding to a distinct category. [4]

For each emotion class, a random 15% of the images were reserved for final testing, entirely held out during model development, and will be referred to as the test set. The remaining images were split into 70% training and 30% validation sets per class, with stratification to ensure that class balance was maintained. The data augmentation techniques mentioned earlier were applied to the training set on the fly during training, while no data augmentation was applied to the validation and test sets to ensure fair performance evaluation.

3.1.2 Model Architectures and Transfer Learning

The model architectures used by the authors of [4] are ResNet50 and VGG-16, implemented with TensorFlow, and MobileNetV1, utilising the Teachable Machine tool [12], [13], [14], [15]. In contrast, this project replaces MobileNetV1 with the more modern MobileNetV2 and utilises PyTorch for development, leveraging its modular design, GPU acceleration, and personal proficiency [16]. These architectures were chosen because of their strong transfer learning performance, pretrained ImageNet weights and prior use in animal emotion recognition. Using ResNet50 and VGG-16 models provides good depth and complexity, while MobileNetV2 offers computational efficiency, thus allowing for a wide range of models to be tested.

Considering the small size of the dataset, one of the only valid options left was to employ transfer learning, taking advantage of the generalised visual features learned by training on a large-scale ImageNet dataset. In deep learning for image analysis, transfer learning is usually achieved by initialising the pretrained model with weights learned on a similar target task, representing simple features standard to all visual tasks, such as edges, textures, shapes, etc. Later, a task-specific approach is chosen to alternate specific layers in the network architecture to suit the needs. Similar to the original work in [4], this project employs various tuning techniques for ResNet50 and VGG-16 to match the architecture details, as explained later in this chapter. Moreover, switching from the Teachable Machine tool to PyTorch enables more controlled tuning of the MobileNetV2 model.

3.1.3 ResNet50 Implementation

Shown in Figure 3.2, ResNet50 is a deep convolutional neural network architecture with 50 layers and residual connections designed to help mitigate the vanishing gradient problem [12]. The original study used ResNet50 implemented in TensorFlow, with a two-stage training approach. First, after initialising the model with pretrained ImageNet weights, all layers were frozen except the final fully connected layer, which was replaced by a new linear layer, acting as a classifier head for a four-class emotion classification. Then, more selected deeper layers were unfrozen for the fine-tuning stage.

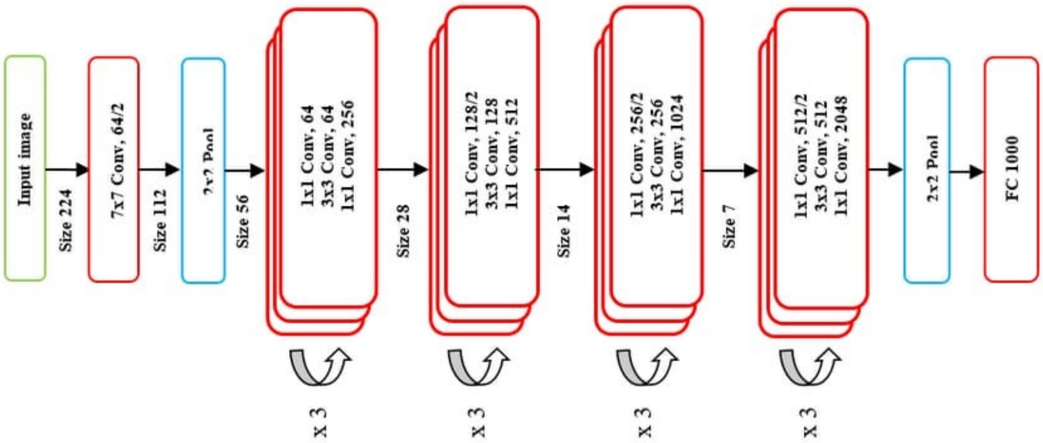


Figure 3.2: Architecture of ResNet50 [17].

Part	Module	Operator	Stride	#Channels	#Blocks
Stem	conv1	Conv7x7	2	64	1
Stem	maxpool	MaxPool3x3	2	64	—
Stage 1	layer1	Bottleneck (1x1–3x3–1x1)	1	256	3
Stage 2	layer2	Bottleneck (1x1–3x3–1x1)	2	512	4
Stage 3	layer3	Bottleneck (1x1–3x3–1x1)	2	1024	6
Stage 4	layer4	Bottleneck (1x1–3x3–1x1)	2	2048	3
Head	avgpool	Global Avg Pool	—	2048	—
Head	fc	FC (2048→num_classes)	—	2048	—

Table 3.1: Stage-level summary of the ResNet-50 backbone. Adapted from [12].

This project is mirroring the same approach, unfreezing layer4 and specifically using Adam optimizer with a learning rate of 1e-4 and training the model for 25 epochs for the head-only training stage [18]. A complete list of the top-level layers of ResNet50 from Torchvision's models library can be found in Table 3.1. Then, additional layers, including layer3 were unfrozen for the fine-tuning stage. A reduced learning rate of 1e-5 was applied for pretrained layers, and the model was further trained for 25 epochs. Early stopping was implemented at both stages if no progress in validation accuracy was seen for 10 epochs.

3.1.4 VGG-16 Implementation

VGG-16 is a uniform deep convolutional neural network architecture, composed of 16 weight layers with sequential 3x3 convolutional layers and periodic max pooling [13], as shown in Figure 3.3. VGG-16 is a uniform deep convolutional neural network architecture, composed of 16 weight layers with sequential 3x3 convolutional layers and max pooling. It is widely used as a feature extractor due to its bottleneck structure, which provides stable intermediate representations. The original paper also utilises this advantage of VGG-16 by only training the classifier head on the target dataset.

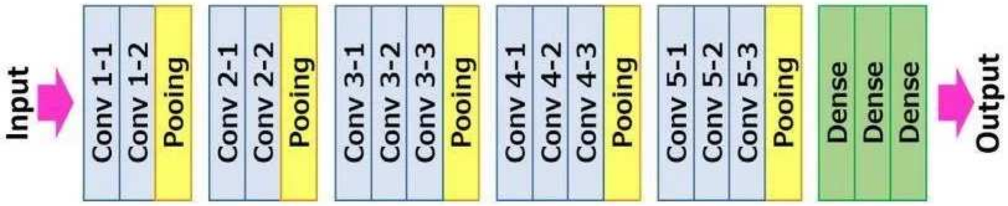


Figure 3.3: Architecture of VGG-16 [19].

Although the original work does not specify how the new head was designed, this project uses an approach proven in transfer learning [20]. The custom classifier consists of fully connected layers, ReLU activations, dropout for regularization and an output layer matching the number of classes. This design is a standard for small dataset tasks, preventing vanishing gradients and overfitting [21], [22]. The architecture of the modified VGG (mVGG) from [20] can be found in Figure 3.4. To suit the input size and the number of output classes for this project, ”Input 256x256x3” was replaced by ”Input 224x224x3” and ”Softmax 5” was replaced by ”Softmax 4”.

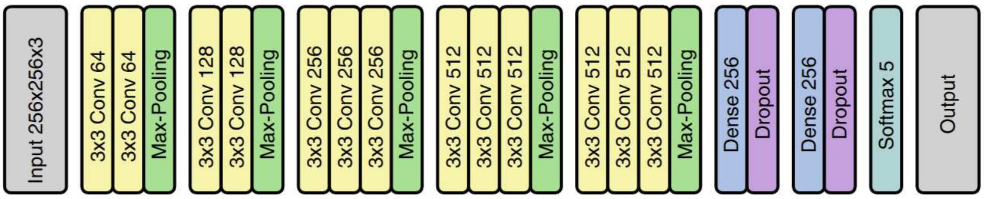


Figure 3.4: Architecture of modified VGG-16 [20].

After passing all the images through the frozen bottleneck, the extracted features were flattened, and the classifier head with a dropout rate of 0.5 was trained for 50 epochs using the Adam optimizer with a learning rate of 1e-4, and early stopping after 10 epochs of no progress in validation accuracy, as in the ResNet50 implementation. Finally, the frozen bottleneck and the trained classifier head were combined into a single model for final test evaluation.

3.1.5 MobileNetV2 Implementation

The last model used by Chávez-Guerrero et al. was MobileNetV1, which was implemented using Google’s Teachable Machine [15]. MobileNet is a lightweight convolutional neural network architecture that utilises depthwise separable convolutions and inverted residuals, as shown in Figure 3.5. This project uses a similar, yet more modern, MobileNetV2 model with PyTorch, allowing for more adjustments compared to the Teachable Machine tool's limited interface.

The authors of the original work used MobileNetV1 for both 4-class and 3-class emotion classification, essentially combining Anxiety and Fear classes from the original dataset into a single class, Anxiety/Fear. This project achieves the same goal, first recreating the 4-class model and then replicating it for the 3-class model with a more balanced class distribution.

For the 4-class model, after initialising the model with ImageNet weights, the final classifier layer was replaced to output 4 classes. Then all the layers were unfrozen for the fine-tuning process, while normalisation and data augmentation were applied where needed. Later, the model was trained for 50 epochs with the Adam optimizer with a learning rate of 1e-4 and cross-entropy loss.

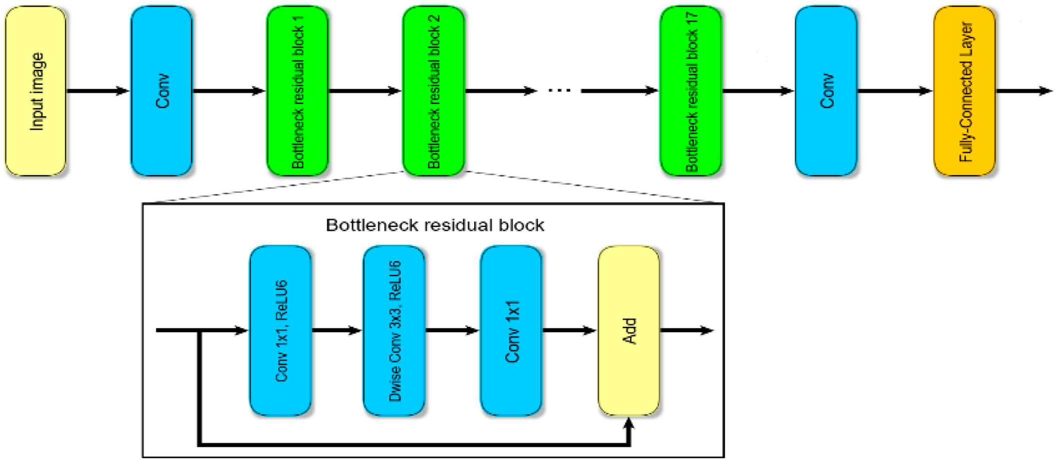

Input image
Conv
Bottleneck residual block 1
Bottleneck residual block 2
...
Bottleneck residual block 17
Conv
Fully-Connected Layer
Bottleneck residual block
Conv 1x1, ReLU6
Dwise Conv 3x3, ReLU6
Conv 1x1
Add

Figure 3.5: Architecture of MobileNetV2 [23].

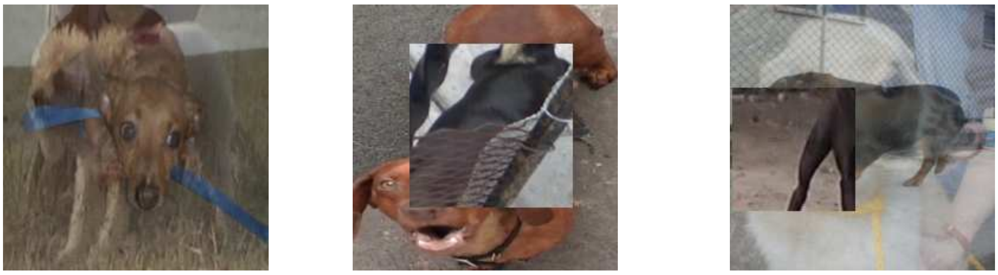
For the 3-class model, Anxiety and Fear classes were combined after the stratified split of the dataset, resulting in more balanced classes with 343 images in the ”Aggressiveness” class, 372 images in the ”Anxiety/Fear” class and 352 images in the ”Neutral” class. The rest of the implementation process remained unchanged, except for replacing the final classifier layer to match three classes. Early stopping was employed for both the 4-class and 3-class models after 10 epochs without any improvement in validation accuracy.

3.2 Enhancements over Chávez-Guerrero et al.

3.2.1 Data Augmentations Experiments

To improve the results and the methodology of the original study, this section utilises several approaches. Firstly, experiments with data augmentations were conducted to observe the effects on the models from Section 3.1 without any essential changes to the methodology or the implementation of the training.[11] Two experiments were conducted, including the CutMix [24] and MixUp [25] techniques in addition to Random Erasing [26], Mosaic Augmentation [27] and Geometric Transformations [11].

After the dataset split as described in Section 3.1, the first experiment uses CutMix, MixUp and the combination of both to generate new images from the training set samples, while keeping the original validation and test sets intact. This prevents data leakage and ensures an unbiased evaluation. To achieve the balance between realism and boundary smoothing, the value of the hyperparameter α for MixUp was set to 0.4, which follows the mid-range setting from the original MixUp paper proven to be effective for image classification [25]. For CutMix, α was set to 1.0 to result in the $\beta(\alpha, \alpha)$ distribution for uniform patch size variety and strong occlusion robustness, matching the original CutMix paper for general classification tasks [24]. Both methods randomly select two images within the same class to use in the data augmentation process to avoid possible label noise issues. The same hyperparameters were used for the sequential combination of both, where MixUp was applied between two images and CutMix was applied between the third image and the result of MixUp.



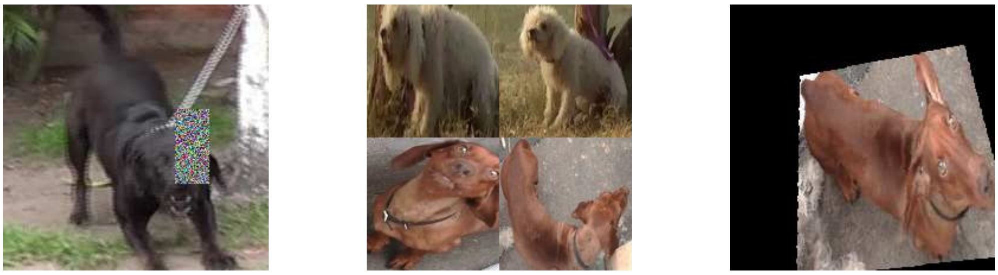
(a) MixUp

(b) CutMix

(c) CutMix and MixUp

Figure 3.6: Augmentation examples from the CutMix and MixUp experiment.

For the second experiment, Random Erasing was applied to all the training set images to generate new images that mimic occlusion or missing parts by erasing a rectangular region of the image and filling it with random values to force the model to rely on global features [26]. The proportion of erasing was set to cover 2%-20% of the image, and the aspect ratio was set to range from 0.3 to 3.3, allowing the occlusions to vary in both length and width. Standard Mosaic augmentation, introduced in YOLOv4, tiles four different images into quadrants by resizing them to half the original resolution [27]. To avoid excessive duplication, the number of mosaics is limited to one-fourth of the number of original images. Thus, ensuring unique image combinations and introducing synthetic background diversity. Geometric Transformations simulate viewpoint changes and minor camera misalignments by applying a sequence of geometric distortions [11]. Random Perspective with a distortion scale of 0.5 allows for a warp of the image with up to 50% shift in corner positions. Random Affine rotates the image within $\pm 15°$ range, shifts it up to $\pm 10\%$ in x and y axes and shears it up to $10°$ in either axis. Such a combination of transformations adds robustness against non-frontal poses, while still avoiding unnatural distortions that might confuse the models.



(a) Random Erasing

(b) Mosaic Augmentation

(c) Geometric Transformations

Figure 3.7: Augmentation examples from the Random Erasing, Mosaic Augmentation and Geometric Transformations experiment.

Each augmentation type, except the combination of MixUp and CutMix, is generated independently from the same training split. All the models see the same original and generated images with data augmentation during training, since all the generated images were created and saved on the disk beforehand.

3.2.2 Model Architecture Experiments

Secondly, experiments with new model architectures were conducted, testing both CNN architectures and a self-supervised Vision Transformer (ViT). The original dataset with the same data split described in Section 3.1 was used for both experiments. Unlike the augmentation experiments in Section 3.2.1, these experiments were conducted on the original dataset using a within-class synthetic minority oversampling technique (SMOTE) for on-the-fly data augmentation, creating synthetic blends with a probability of 0.7. SMOTE-style augmentation was used only on the training split [28].

EfficientNetV2-M architecture, which integrates compound scaling with improved building blocks and is pretrained on ImageNet-21k, was chosen as the first candidate for the CNN architecture experiment [29]. The classifier head for the model was replaced with a new one with a dropout layer and a linear output layer for the target classes. Unlike the VGG-16 in Section 3.1.4, EfficientNetV2-M already has drop path structural regularization, hence it does not require a strong dropout regularization [30]. Considering that and a large-scale pretraining, the dropout rate was reduced to 0.1, while the drop path rate was set to 0.3. Moreover, class-weighted cross-entropy was used to compensate for the dataset imbalance by penalising incorrect predictions in under-represented classes more strictly. The label smoothing hyperparameter ϵ was set to 0.1 to prevent the model from making overly confident predictions [31].

Part	Module	Operator	Stride	#Channels	#Blocks
Stem	conv_stem	Conv3x3	2	24	1
Stage 0	blocks[0]	Fused-MBConv, k3x3	1	24	3
Stage 1	blocks[1]	Fused-MBConv, k3x3	2	48	5
Stage 2	blocks[2]	Fused-MBConv, k3x3	2	80	5
Stage 3	blocks[3]	MBConv, k3x3, SE	2	160	7
Stage 4	blocks[4]	MBConv, k3x3, SE	1	176	14
Stage 5	blocks[5]	MBConv, k3x3, SE	2	304	18
Stage 6	blocks[6]	MBConv, k3x3, SE	1	512	5
Head	conv_head	Conv1x1	1	1280	—
Head	classifier	FC (1280→num_classes)	—	1280	—

Table 3.2: Stage-level summary of the EfficientNetV2-M backbone. Adapted from [29].

The training was performed in 3 stages with AdamW optimizer and weight decay of 0.1 [32]. In the first stage, the learning rate was set to 1e-4, all pretrained backbone parameters were frozen, and only the new classifier was trained for 10 epochs. For the second stage, the learning rate was reduced to 5e-5, and an additional three blocks were unfrozen. Those included blocks[4], blocks[5] and blocks[6] from Table 3.2. Moreover, cosine annealing was introduced for a gradual decrease of the learning rate following a cosine curve, and the model was trained for an additional 10 epochs [33]. In the third stage, the entire network was unfrozen, the learning rate was further decreased to 1e-5, and the model was trained for 30 more epochs with a cosine annealing learning rate scheduler.

The second choice of CNN architecture fell on ConvNeXt-Base as a modern ResNet reinterpretation, taking inspiration from the transformer design, including large kernel sizes, inverted bottlenecks, and layer scaling [34]. Similar to the EfficientNetV2-M training, the AdamW optimizer, cosine annealing learning rate scheduler and the class-weighted cross-entropy were used with the same hyperparameter settings. The training process was also performed in 3 stages for the same number of epochs, starting with unfreezing the classifier head, which was created to match the number of classes. Then stages[1], stages[2] and stages[3] shown in Table 3.3 were unfrozen for the second stage of training. Finally, the entire network was unfrozen in the third stage of training.

Part	Module	Operator	Stride	#Channels	#Blocks
Stem	stem	Conv4x4	4	128	1
Stage 0	stages[0]	ConvNeXtBlock, k7x7	1	128	3
Stage 1	stages[1]	ConvNeXtBlock, k7x7	2	256	3
Stage 2	stages[2]	ConvNeXtBlock, k7x7	2	512	27
Stage 3	stages[3]	ConvNeXtBlock, k7x7	2	1024	3
Head	head	FC (1024→num_classes)	—	1024	1

Table 3.3: Stage-level summary of the ConvNeXt-Base backbone. Adapted from [34].

DINOv2 ViT-S/14 was chosen for the second experiment to find out whether a transformer-based backbone can offer advantages over CNN architectures [35]. The training was also similarly divided into 3 stages, with the first stage unfreezing only the new classifier head consisting of a LayerNorm with ϵ set to 1e-6 and a linear layer for 4 output classes [36]. Since the input image size is 224×224, the token size is 16×16 for a patch size of 14 fixed for the DINOv2 ViT-S/14. Blocks b9, b10 and b11 shown in Table 3.4 were unfrozen in the second stage, and the rest of the blocks were unfrozen in the third stage. For a fair comparison with EfficientNetV2-M and ConvNeXt, the learning rate, loss criterion, learning rate scheduler and the optimizer settings were unchanged.

Block	d_{model}	MHSA (qkv / proj)	MLP (hidden)	Notes
Patch Embedding	384	—	—	Conv 14×14, stride 14
b0	384	384→1152 / 384	384→1536→384	LS; DP= 0
b1	384	384→1152 / 384	384→1536→384	LS; DP= 0
b2	384	384→1152 / 384	384→1536→384	LS; DP= 0
b3	384	384→1152 / 384	384→1536→384	LS; DP= 0
b4	384	384→1152 / 384	384→1536→384	LS; DP= 0
b5	384	384→1152 / 384	384→1536→384	LS; DP= 0
b6	384	384→1152 / 384	384→1536→384	LS; DP= 0
b7	384	384→1152 / 384	384→1536→384	LS; DP= 0
b8	384	384→1152 / 384	384→1536→384	LS; DP= 0
b9	384	384→1152 / 384	384→1536→384	LS; DP= 0
b10	384	384→1152 / 384	384→1536→384	LS; DP= 0
b11	384	384→1152 / 384	384→1536→384	LS; DP= 0
Norm	384	—	—	LayerNorm
Head	384	—	—	Identity head

Table 3.4: Block-level summary for DINOv2 ViT-S/14. Each encoder block has the sequence: **LN** → **MHSA** → residual add with **LS** and **DP** → **LN** → **MLP** (fc1 → GELU → fc2) → residual add with **LS** and **DP**. **Acronyms:** MHSA - multi-head self-attention; qkv - query/key/value linear projection; proj - attention output projection; MLP - two-layer feed-forward with GELU; LS - LayerScale; DP - DropPath (disabled here, i.e., acts as Identity); LN - LayerNorm. Adapted from [35].

3.2.3 Combined Dataset

Lastly, experiments with a new dataset, combined from 3 dog emotion datasets, were conducted to find out how the models would perform on a larger and more class-balanced dataset. In addition to the original dataset, a dataset from [8], available upon request to the authors of the paper, and a Dog Emotion Dataset from Kaggle, which is open-access on

Kaggle [37]. They will be referred to as the DEBiW dataset and the KaggleDogEmotion dataset.

The DEBiW dataset is a web collection of nearly 16,000 dog images, where each of those was annotated by up to 7 trained volunteers, providing a categorical label with valence and arousal on a 1-5 scale. Initially, one of the five labels —Aggression, Anxiety, Contentment, Fear, or None/No Dog— was assigned to each image by each annotator. For this project, None/NoDog images were discarded. The images, for which all annotators agreed on the label, were assigned a definite label corresponding to the emotion they evoked. Based on the top emotion vote count, the images with all-but-one agreement, for which only one annotator's vote differed from the others, were assigned one of the following labels: mostly_Aggression, mostly_Anxiety, mostly_Contentment or mostly_Fear. The remaining images were assigned to the ambiguous category. Before downloading, the dataset was further filtered to discard all images without a definite filter and with fewer than two annotator votes. Moreover, the images were run through the YOLOv5 model to detect the dogs and skip those that do not contain dogs [38]. Finally, the largest dog box area was cropped for each image containing dogs after YOLOv5 detection to further enhance the dataset quality. Thus, the number of images qualifying for the experiment drastically dropped, resulting in 316 images for the Aggression class, 147 for the Anxiety class, 792 for the Contentment class and 272 for the Fear class.

The KaggleDogEmotion dataset consists of 4 emotion classes: angry, happy, relaxed and sad. Each of the classes contains 1000 images, making the dataset extremely class-balanced. No changes were made to the dataset for this project, as it was of high quality and usability.

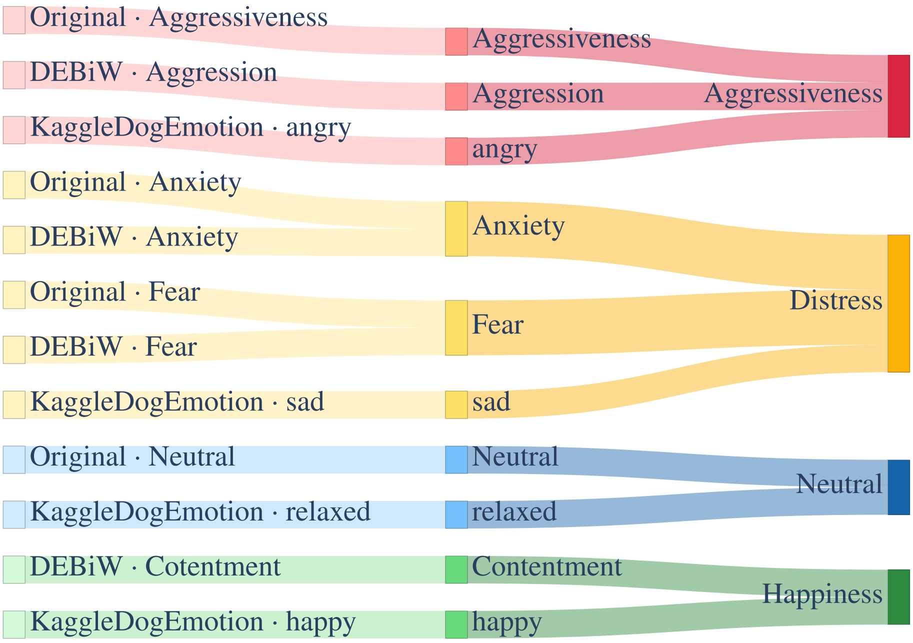


Figure 3.8: Visualisation of the class merging.

The "Aggressiveness" class from the original dataset, the "Aggression" class from the DEBiW dataset, and the "angry" class from the KaggleDogEmotion dataset were combined into the combined "Aggressiveness" class and saved as subclasses. Since both the original dataset and the DEBiW dataset contain "Anxiety" and "Fear" classes, they were combined into the subclasses with the same names. Together with the "sad" subclass, taken from the Kaggle-

DogEmotion dataset, these compromised the ”Distress” class of the combined dataset. The ”Neutral” class of the original dataset and the ”relaxed” class of the KaggleDogEmotion dataset, as well as the ”Contentment” class of the DEBiW dataset and the ”happy” class of the KaggleDogEmotion dataset, were respectively used as subclasses for the ”Neutral” and ”Happiness” classes of the combined dataset. The visual description of class merging can be found in Figure 3.8. The final combined dataset comprises 1662 images in the ”Aggressiveness” class, 1352 images in the ”Neutral” class, 1792 images in the ”Happiness” class and 1790 images in the ”Distress” class.

3.2.4 Combined Dataset Experiments

For the combined dataset experiments, all 6 models mentioned in Section 3.1.2 and Section 3.2.2 were used with a unified training pipeline. First, 10% of images from each subclass of the final dataset described in Section 3.2.3 were held out for the test set, and then the remaining images were split into the training and validation sets with an 80/20 split with each subclass. The pipeline was essentially unchanged compared to the one used for the models in Section 3.2.2, utilising the same 3-stage training with 10 epochs for each stage, as well as the same learning rates, weight decay, and cosine annealing learning rate scheduler. Despite using simple data augmentations, due to a larger and more balanced dataset, the SMOTE-style augmentations were omitted. Moreover, the loss criterion was switched back to a simple cross-entropy without label smoothing. Per stage layer and module unfreezing for each model is shown in Table 3.5.

Model	**Stage 1**	**Stage 2**	**Stage 3**
ResNet-50	layer4 + classifier head	layer3 + layer4 + classifier head	Entire network
VGG-16	classifier head	last conv block (conv5_x) + classifier head	Entire network
MobileNetV2	classifier head	last 3 inverted residual blocks + classifier head	Entire network
EfficientNetV2-M	classifier head	last 3 blocks + classifier head	Entire network
ConvNeXt-Base	classifier head	last 3 stages + classifier head	Entire network
DINOv2 ViT-S/14	classifier head (LN+Linear)	last 3 transformer blocks + classifier head	Entire network

Table 3.5: Layers/modules unfrozen at each training stage. LN: LayerNorm.

3.3 Evaluation and Interpretability

Metrics For evaluation and interpretability, a unified methodology was employed across all models and experiments in this project. Both overall performance on the held-out test set, as well as accuracy, macro-F1, weighted-F1, and per-class performance, including precision, recall, F1 score, support, and per-class accuracy, were tracked alongside the best and worst accuracies for training and validation [39]. Models were evaluated in the evaluation mode without test-time augmentation. Predictions were obtained by *argmax* over logits.

Accuracy Overall fraction of correct predictions.

$$\mathrm{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[\hat{y}_i = y_i].$$

Precision (per class) Of all samples predicted as class c, the proportion that are truly c (low false positives).

$$\mathrm{Prec}_c = \frac{TP_c}{TP_c + FP_c}.$$

Recall (per class) Of all true samples of class c, the proportion correctly found (low false negatives).

$$\mathrm{Rec}_c = \frac{TP_c}{TP_c + FN_c}.$$

F1 (per class) Harmonic mean of precision and recall for class c.

$$\mathrm{F1}_c = \frac{2\,\mathrm{Prec}_c\,\mathrm{Rec}_c}{\mathrm{Prec}_c + \mathrm{Rec}_c} \;=\; \frac{2TP_c}{2TP_c + FP_c + FN_c}.$$

Macro-F1 Unweighted mean of per-class F1; treats all classes equally.

$$\mathrm{MacroF1} = \frac{1}{C}\sum_{c=1}^{C} \mathrm{F1}_c.$$

Weighted-F1 Mean of per-class F1 weighted by class support; reflects overall class frequencies.

$$\mathrm{WeightedF1} = \sum_{c=1}^{C} \frac{n_c}{N}\,\mathrm{F1}_c.$$

Support (per class) Number of true test samples in class c.

$$\mathrm{Support}_c = n_c.$$

Note. The per-class accuracy reported is the row-wise recall derived from the confusion matrix.

Confusion Matrix Additionally, confusion matrices were used to provide a comprehensive visual summary of errors, reported as raw counts with cell annotations. Multiclass errors are summarised with a confusion matrix $C \in \mathbb{N}^{C\times C}$ on the held-out test set, using the convention *rows = true class*, *columns = predicted class*. Element $C_{r,c}$ counts how many samples of true class r were predicted as c.

$$C_{r,c} \;=\; \sum_{i=1}^{N} \mathbf{1}[\,y_i = r\,]\,\mathbf{1}[\,\hat{y}_i = c\,], \qquad r, c \in \{1, \ldots, C\}.$$

Per-class error counts are recovered from C:

$$\mathrm{TP}_c = C_{c,c}, \quad \mathrm{FN}_c = \sum_{j=1}^{C} C_{c,j} - C_{c,c}, \quad \mathrm{FP}_c = \sum_{i=1}^{C} C_{i,c} - C_{c,c}.$$

Overall accuracy and common aggregates follow directly:

$$\mathrm{Acc} \;=\; \frac{\mathrm{trace}(C)}{\sum_{r,c} C_{r,c}}.$$

EigenCAM Similar to [9], a gradient-free, class-agnostic EigenCAM saliency method was used for the prediction interpretability and to spot background bias or wrong cues [40]. EigenCAM uses the first principal component (PC1) to turn a layer activation tensor into a single heatmap.

For CNNs in this project, EigenCAM hooks the last high-level convolutional output of the automatically detected `Conv2D` layer with a spatial grid encoding semantic parts. Then, the information across channels is combined to produce a single score per spatial location, which is later unsampled and overlaid on the image. Whereas for ViT, it hooks the token embeddings from the last transformer block, drops the classification token and keeps only patch tokens. Then, information across feature dimensions is combined to produce one score per patch and is reshaped to the patch grid. Unsample and overlay processes are identical to those for CNNs. Specific layers hooked by EigenCAMs for different models can be found in Table 3.6.

Model	**Hook target**	**Layer**
ResNet–50	`layer4`	Final conv stage (conv5_x), pre–global pooling.
VGG–16	`features[24:]`	Last conv block (conv5), pre–avgpool.
MobileNetV2	`features[-1]`	Final ConvBNReLU / feature map before classifier.
EfficientNetV2–M	`blocks[-1]`	Last (fused-)MBConv stage output, pre–head.
ConvNeXt–Base	`stages[-1]`	Last ConvNeXt stage output, pre–head.
DINOv2 ViT–S/14	`blocks[-1]`	Last transformer block tokens.

Table 3.6: Hooked layers used to extract activations for EigenCAM across the six models.

For each experiment, the first correctly and the first incorrectly predicted test images per class are selected, and the corresponding EigenCAM overlays are generated. These overlays are intended to aid in interpreting the predictions, rather than relying solely on metrics and numbers.

Chapter 4: Results

4.1 Results of the reproduction of the original work by Chávez-Guerrero et al.

This chapter showcases the results of both the reproduction and the enhancement experiments for the original work by Chávez-Guerrero et al. To start with, both the general and per-class metric results for all 4-class models and the 3-class MobileNetV2 model are shown in Table 4.1.

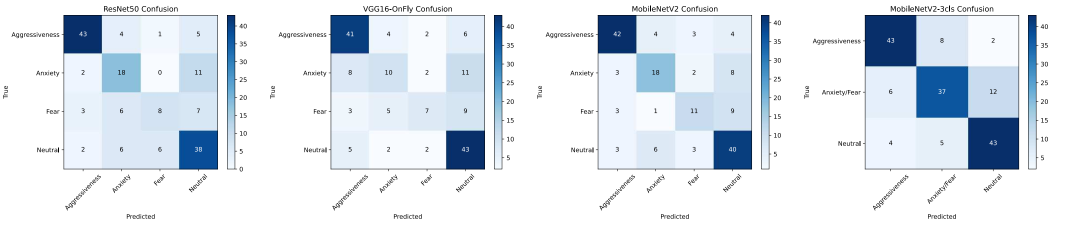
Model	Test Acc	Macro F1	Weighted F1	Best Val	Worst Val	Best Train
ResNet50	0.6687	0.6179	0.6640	0.7184	0.5993	0.9984
VGG-16	0.6312	0.5548	0.6092	0.6643	0.5343	0.9236
MobileNetV2	0.6937	0.6568	0.6906	0.7220	0.5451	1.0000
MobileNetV2 (3-cls)	**0.7688**	**0.7684**	**0.7674**	**0.7798**	0.6570	0.9984

Model	Class	Prec	Recall	F1	Support	Acc
ResNet50 (4-cls)	Aggressiveness	0.8600	0.8113	**0.8350**	53	0.8113
	Anxiety	0.5294	0.5806	0.5538	31	0.5806
	Fear	0.5333	0.3333	0.4103	24	0.3333
	Neutral	0.6230	0.7308	0.6726	52	0.7308
VGG-16 (4-cls)	Aggressiveness	0.7193	0.7736	0.7455	53	0.7736
	Anxiety	0.4762	0.3226	0.3846	31	0.3226
	Fear	0.5385	0.2917	0.3784	24	0.2917
	Neutral	0.6232	0.8269	0.7107	52	0.8269
MobileNetV2 (4-cls)	Aggressiveness	0.8235	0.7925	0.8077	53	0.7925
	Anxiety	0.6207	0.5806	**0.6000**	31	0.5806
	Fear	0.5789	0.4583	**0.5116**	24	0.4583
	Neutral	0.6557	0.7692	0.7080	52	0.7692
MobileNetV2 (3-cls)	Aggressiveness	0.8113	0.8113	0.8113	53	0.8113
	Anxiety/Fear	0.7400	0.6727	0.7048	55	0.6727
	Neutral	0.7544	0.8269	**0.7890**	52	0.8269

Table 4.1: Reproduction results for 4-class and 3-class models.

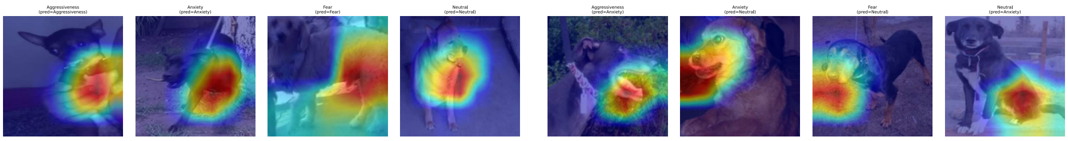
The PyTorch implementation of ResNet50 in this project achieved a test accuracy of 0.6687 compared to 0.3002 in the original study [4]. VGG-16 also showed a significant improvement in test accuracy, from 0.3243 to 0.6312, while MobileNet's accuracy remained almost unchanged, increasing from 0.6917 to 0.6937. Although the final test accuracy for the 3-class model is not mentioned in the original work, per-class accuracies can still be compared to assess the improvement of MobileNetV2. The results for the Aggressiveness class improved from 71.1% by 10%; for the combined class of Anxiety and Fear, only by 3.1%; and by 18.6% for the Neutral class, resulting in 82.7% accuracy. Overall, all the models, except the 4-class MobileNetV2, noticeably outperform those from the original study.

Figure 4.1 shows a more detailed performance of each model on the test set. F1 scores clearly show that all 4-class models struggle with Anxiety and Fear classes, and the 3-class model partially solved this issue by combining them into a single class.



(a) ResNet50 (b) VGG-16 (c) MobileNetV2 (d) 3-class MobileNetV2

Figure 4.1: Confusion matrices on the test set for reproduced models.

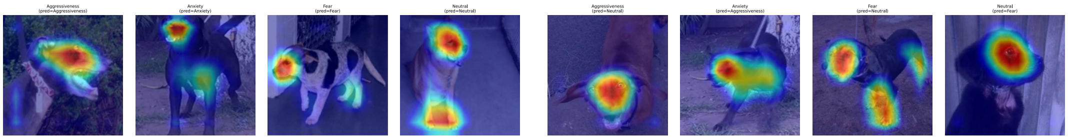


(a) Correct predictions (b) Incorrect predictions

Figure 4.2: EigenCAM overlays — ResNet50 reproduction.

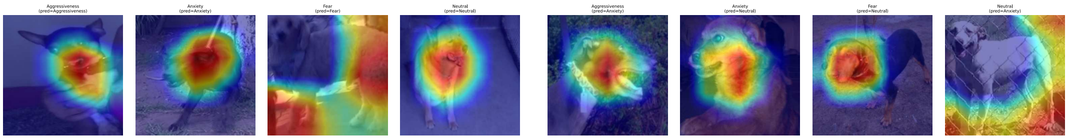
Referring to Figure 4.2, ResNet50 mainly focuses on the front chest in the correctly predicted images and on the face in the incorrectly predicted ones. Moreover, incorrect examples suggest that the attention drifts to non-diagnostic regions or background objects. For instance, the heatmap for the Fear class in Figure 4.2a is diffuse and slightly off-target despite the correct prediction. This pattern further supports a low recall score for the Fear class in Table 4.1.

In contrast, as can be seen from Figure 4.3, VGG-16 generates very compact face-centric, sometimes multiple, hotspots with little attention to the body and attention drift to legs in some cases.



(a) Correct predictions (b) Incorrect predictions

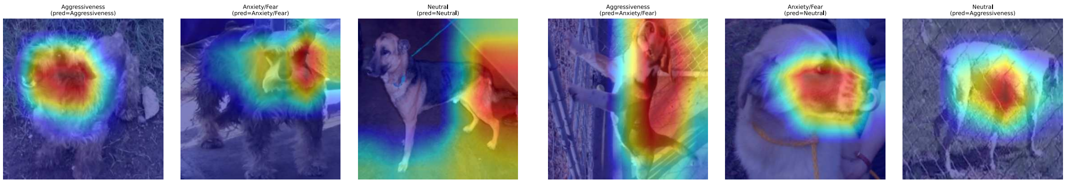
Figure 4.3: EigenCAM overlays — VGG-16 reproduction.



(a) Correct predictions (b) Incorrect predictions

Figure 4.4: EigenCAM overlays — 4-class MobileNetV2 reproduction.

MobileNetV2 focuses both on the face and the torso, depending on the image, as shown in Figure 4.4. More noticeably, sometimes the activation undergoes a substantial shift towards the background and textures. For example, the model completely ignores the dog in the Neutral class example in Figure 4.4b.



(a) Correct predictions (b) Incorrect predictions

Figure 4.5: EigenCAM overlays — 3-class MobileNetV2 reproduction.

In Figure 4.5, EigenCAM generates more stable hotspots covering the head and torso with fewer distractions to the background for the 3-class model. Although, as shown in Figure 4.5a, the activations still were diffused for the Neutral class example.

4.2 Results of Data Augmentation Experiments

4.2.1 CutMix and MixUp Experiment Results

As the results of the experiment described in Section 3.2.1 show in Table 4.2, CutMix and MixUp augmentation helped ResNet50 and MobileNetV2 achieve slight improvements in their results with +0.0188 and +0.0313 test accuracy increases, respectively. Meanwhile, despite the +0.1437 F1 score increase for the Anxiety class, the overall test accuracy for VGG-16 dropped due to the worse per-class performance for the remaining three classes. Macro-F1, Weighted-F1 and per-class F1 scores further prove that the only model clearly benefiting from the experiment is MobileNetV2, performing noticeably better on the problematic Anxiety and Fear classes.

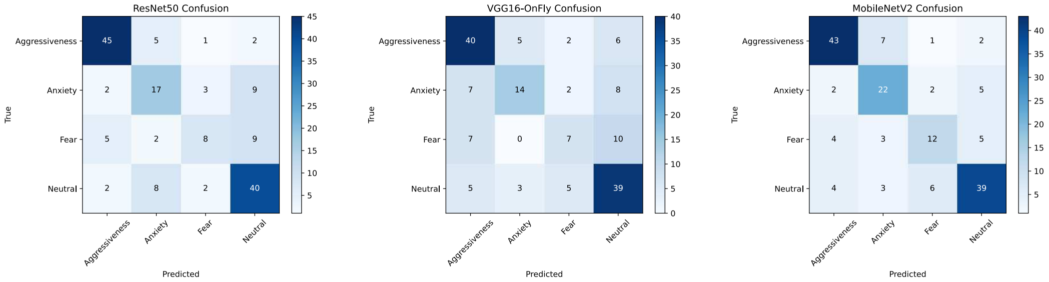
Model	Test Acc	Macro F1	Weighted F1	Best Val	Worst Val	Best Train
ResNet50	0.6875	0.6290	0.6785	0.7581	0.4079	0.9994
VGG-16	0.6250	0.5677	0.6119	0.7365	0.6173	1.0000
MobileNetV2	**0.7250**	**0.6922**	**0.7240**	**0.7726**	0.6318	0.9994

Model	Class	Prec	Recall	F1	Support	Acc
ResNet50	Aggressiveness	0.8333	0.8491	**0.8411**	53	0.8491
	Anxiety	0.5312	0.5484	0.5397	31	0.5484
	Fear	0.5714	0.3333	0.4211	24	0.3333
	Neutral	0.6667	0.7692	0.7143	52	0.7692
VGG-16	Aggressiveness	0.6780	0.7547	0.7143	53	0.7547
	Anxiety	0.6364	0.4516	0.5283	31	0.4516
	Fear	0.4375	0.2917	0.3500	24	0.2917
	Neutral	0.6190	0.7500	0.6783	52	0.7500
MobileNetV2	Aggressiveness	0.8113	0.8113	0.8113	53	0.8113
	Anxiety	0.6286	0.7097	**0.6667**	31	0.7097
	Fear	0.5714	0.5000	**0.5333**	24	0.5000
	Neutral	0.7647	0.7500	**0.7573**	52	0.7500

Table 4.2: CutMix and MixUp experiment results.

Diagonal elements of the confusion matrices in Figure 4.6 show that all three models perform well on Agressiveness and Neutral classes. However, they still struggle with Anxiety and Fear, similarly to the results in Section 4.1. Overall, Fear remained the most challenging class for all models, which can be explained by its low support value.

Figure 4.7 shows that ResNet50 predictions were mainly chest-centric with background pull, similar to Section 4.1. Overall, visual patterns remained mostly unchanged with a bit broader heatmaps.

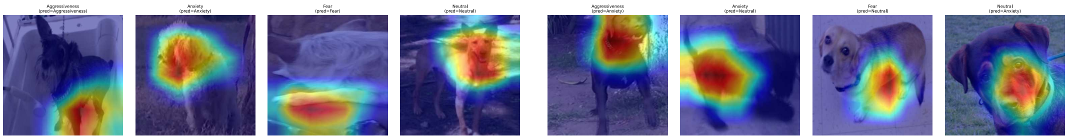


(a) ResNet50

(b) VGG-16

(c) MobileNetV2

Figure 4.6: Confusion matrices for the CutMix and MixUp experiment.

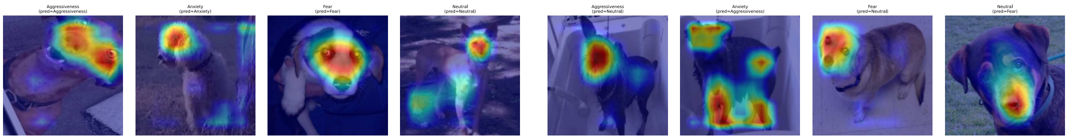


(a) Correct

(b) Incorrect

Figure 4.7: EigenCAM overlays — ResNet50 with CutMix and MixUp.

For VGG-16, as can be seen in Figure 4.8, the predictions remain face-centric with occasional torso coverage. Multiple small blob patterns and leg drift in some examples indicate the attention fragmentation matching mixed class-wise outcomes.

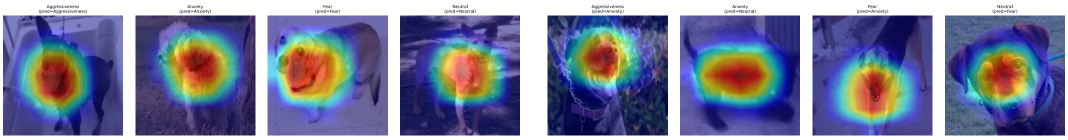


(a) Correct

(b) Incorrect

Figure 4.8: EigenCAM overlays — VGG-16 with CutMix and MixUp.

MobileNetV2 EigenCAM heatmaps, similarly to the accuracy and per-class performance, improved the most with more stable head and torso coverage, as shown in Figure 4.9. Moreover, the background pulls significantly weakened with fewer dog detection failures compared to Section 4.1.



(a) Correct

(b) Incorrect

Figure 4.9: EigenCAM overlays — MobileNetV2 with CutMix and MixUp.

4.2.2 Random Erasing, Mosaic Augmentation and Geometric Transformations Experiment Results

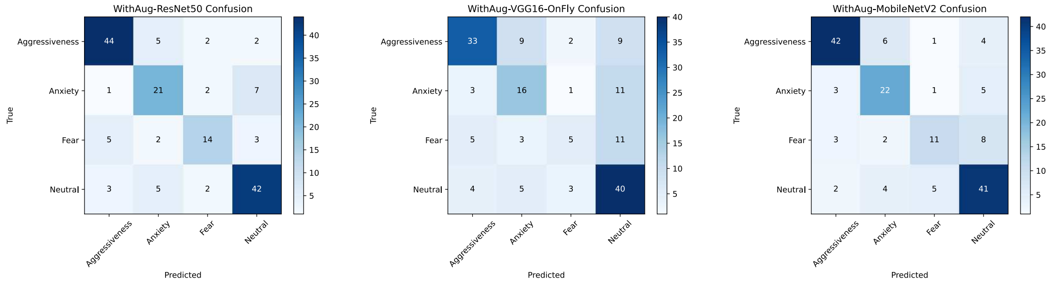
As shown in Table 4.3, ResNet50 results significantly improved in the second data augmentation experiment described in the second experiment in Section 3.2.1 compared to the original work reproduction. Overall test accuracy increased by +0.0875, and the weighted-F1 score increased by +0.0912, proving the positive influence of the second experiment's data augmentation methods on ResNet50, in contrast to relatively weaker improvements of MobileNetV2 and much worse performance of VGG-16. Per-class F1 scores also indicate that ResNet50 performs the best on all classes except the Anxiety class, where it falls slightly behind MobileNetV2.

Model	Test Acc	Macro F1	Weighted F1	Best Val	Worst Val	Best Train
ResNet50	**0.7562**	**0.7288**	**0.7552**	**0.7617**	0.4657	0.9986
VGG-16	0.5875	0.5274	0.5742	0.7256	0.4946	0.9697
MobileNetV2	0.7250	0.6904	0.7221	**0.7617**	0.6029	0.9981

Model	Class	Prec	Recall	F1	Support	Acc
ResNet50	Aggressiveness	0.8302	0.8302	**0.8302**	53	0.8302
	Anxiety	0.6364	0.6774	0.6562	31	0.6774
	Fear	0.7000	0.5833	**0.6364**	24	0.5833
	Neutral	0.7778	0.8077	**0.7925**	52	0.8077
VGG-16	Aggressiveness	0.7333	0.6226	0.6735	53	0.6226
	Anxiety	0.4848	0.5161	0.5000	31	0.5161
	Fear	0.4545	0.2083	0.2857	24	0.2083
	Neutral	0.5634	0.7692	0.6504	52	0.7692
MobileNetV2	Aggressiveness	0.8400	0.7925	0.8155	53	0.7925
	Anxiety	0.6471	0.7097	**0.6769**	31	0.7097
	Fear	0.6111	0.4583	0.5238	24	0.4583
	Neutral	0.7069	0.7885	0.7455	52	0.7885

Table 4.3: Random Erasing, Mosaic Augmentation and Geometric transformations experiment results.

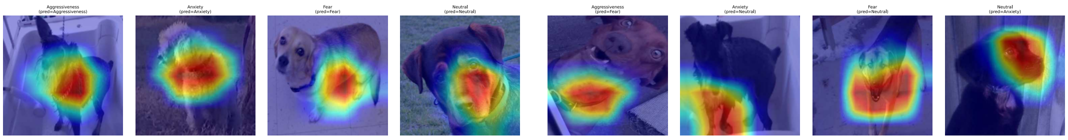
Confusion matrices in Figure 4.10 indicate that ResNet50 was the most balanced overall, and MobileNetV2 often confused Fear for Neutral. At the same time, VGG-16 clearly had a bias towards the Neutral class, as many images were misclassified as Neutral.



(a) ResNet50 (b) VGG-16 (c) MobileNetV2

Figure 4.10: Confusion matrices for the Random Erasing, Mosaic Augmentation and Geometric Transformations experiment.

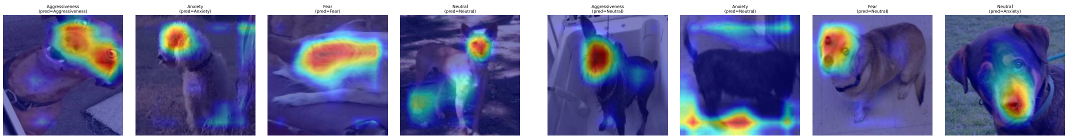
With new augmentation, as can be seen in Figure 4.11, ResNet50 hotspots widened from mainly purely chest to chest and neck. Background pull was reduced compared to Figure 4.2, and heatmap localisation became more stable. Figure 4.12 shows that the attention pattern for VGG-16 was largely unchanged with small, tight multi-blob drift to legs, ground and background, aligning with overall weak performance. And for MobileNetV2, in Fugere 4.13, more coherent head and torso coverage can be seen with occasional but less severe background bias. MobileNetV2 also shows better class-relevant focus in Figure 4.13a compared to other models, for example, focusing on the teeth for the Aggressiveness class and on the eyes for the Fear class.



(a) Correct

(b) Incorrect

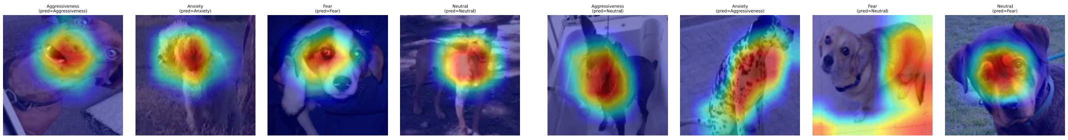
Figure 4.11: EigenCAM overlays — ResNet50 with Random Erasing, Mosaic, and Geometric Transformations.



(a) Correct

(b) Incorrect

Figure 4.12: EigenCAM overlays — VGG-16 with Random Erasing, Mosaic, and Geometric Transformations.



(a) Correct

(b) Incorrect

Figure 4.13: EigenCAM overlays — MobileNetV2 with Random Erasing, Mosaic, and Geometric Transformations.

To conclude the data augmentation experiments, ResNet50 and MobileNetV2 gained clear accuracy boosts from both experiments, while VGG-16 remained the weakest model with even worse test accuracy. The most common confusions were Fear and Anxiety, classified as Neutral, which were specifically distinguishable in VGG-16. Overall, augmentations did not remove core failure modes, despite architecture-dependent performance improvements.

4.3 Results of Model Architecture Experiments

As can be concluded from Table 4.4, in the model architecture experiments described in Section 3.2.2, all three models performed nearly at the same level, with EfficientNetV2-M being the best in test accuracy, macro-F1 and weighted-F1 scores by a very small margin. However, at the same time, EfficientNetV2-M also showed a relatively lower best training accuracy, indicating slight underfitting to the training data.

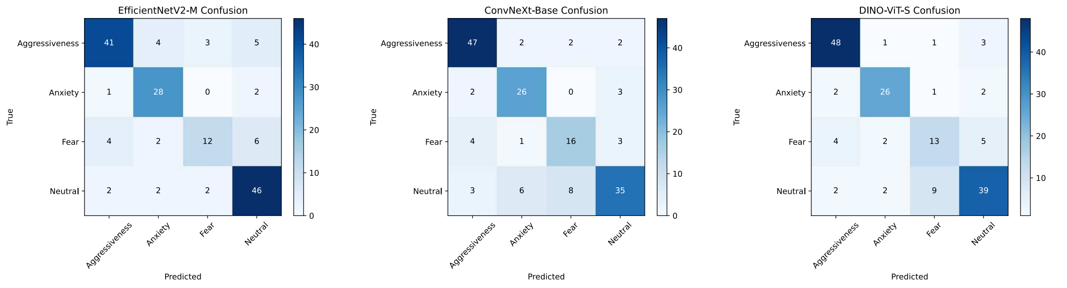
In per-class performance, EfficientNetV2-M performed the best on Neutral and Anxiety classes with F1 scores of 0.8288 and 0.8358, respectively. DINOv2 ViT-S/14 scored the best on the Aggressiveness class, with a performance comparable to EfficientNetV2-M on the Anxiety class. Noticeably, the ConvNeXt-Base model outperformed the other two models on the most problematic Fear class with an F1 score of 0.6400, while also just falling behind DINOv2 ViT-S/14 on the Aggressiveness class.

Model	Test Acc	Macro F1	Weighted F1	Best Val	Worst Val	Best Train
EfficientNetV2-M	**0.7937**	**0.7655**	**0.7881**	0.7220	0.3466	0.9267
ConvNeXt-Base	0.7750	0.7568	0.7738	0.7437	0.2816	0.9969
DINOv2 ViT-S/14	0.7875	0.7583	0.7865	**0.7942**	0.2527	0.9953

Model	Class	Prec	Recall	F1	Support	Acc
EfficientNetV2-M	Aggressiveness	0.8542	0.7736	0.8119	53	0.7736
	Anxiety	0.7778	0.9032	0.8358	31	0.9032
	Fear	0.7059	0.5000	0.5854	24	0.5000
	Neutral	0.7797	0.8846	**0.8288**	52	0.8846
ConvNeXt-Base	Aggressiveness	0.8393	0.8868	0.8624	53	0.8868
	Anxiety	0.7429	0.8387	0.7879	31	0.8387
	Fear	0.6154	0.6667	**0.6400**	24	0.6667
	Neutral	0.8140	0.6731	0.7368	52	0.6731
DINOv2 ViT-S/14	Aggressiveness	0.8571	0.9057	**0.8807**	53	0.9057
	Anxiety	0.8387	0.8387	**0.8387**	31	0.8387
	Fear	0.5417	0.5417	0.5417	24	0.5417
	Neutral	0.7959	0.7500	0.7723	52	0.7500

Table 4.4: Model architecture experiment results.

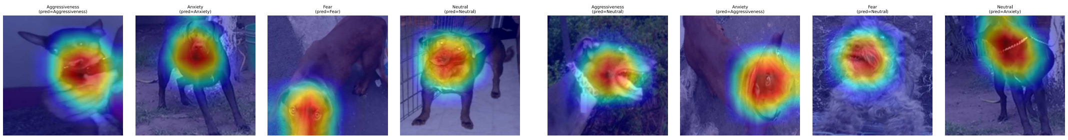
Confusion matrices in Figure 4.14 evince that new model architectures demonstrated a better performance on the Anxiety class, although still somewhat struggled with the Fear class. Despite significantly reducing the Neutral class bias, main confusions still occurred between the Neutral and Fear classes. Overall, EfficientNetV2-M proved to be the most balanced, ConvNeXt-Base and DINOv2 ViT-S/14 underperformed on the Neutral class.



(a) EfficientNetV2-M (b) ConvNeXt-Base (c) DINO-ViT-S

Figure 4.14: Confusion matrices for the model architecture experiment.

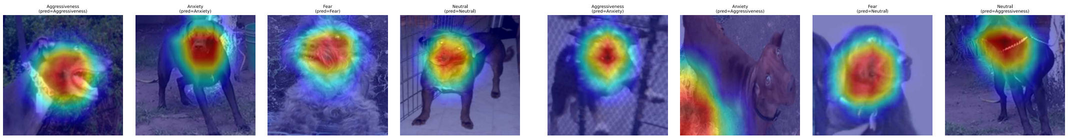
As can be seen in Figure 4.15, the EigenCAM heatmaps for EfficientNetV2 are compact, with single hotspots centred on the head in both correctly and incorrectly predicted images. Such stable localisation matches the model's overall strong performance.



(a) Correct (b) Incorrect

Figure 4.15: EigenCAM overlays — EfficientNetV2-M.

Figure 4.16 demonstrates that ConvNeXt-Base EigenCAM overlays produced largely similar heatmaps with occasional yet strong background attention pull. Unstable behaviours can be clearly seen in Figure 4.16b, where the model was able to ignore the fence in the Aggressiveness class example and was completely distracted by the background in the Anxiety class example.

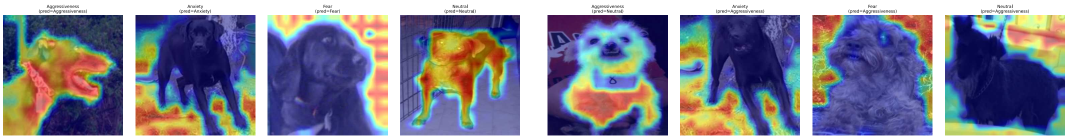


(a) Correct

(b) Incorrect

Figure 4.16: EigenCAM overlays — ConvNeXt-Base.

DINOv2 ViT-S/14, as can be seen in Figure 4.17, produced the most unstable EigenCAM overlays, contradicting strong performance and highlighting architectural differences between CNNs and ViTs. In some correctly predicted images, the model was able to lock onto the dog and its heatmaps covered the whole body, although including the leash in the Aggressiveness class example in Figure 4.17a. However, in most cases, where the background dominates the image, the saliency spreads to a large background area, totally ignoring the image.



(a) Correct

(b) Incorrect

Figure 4.17: EigenCAM overlays — DINOv2 ViT-S/14.

Model architecture experiment results show that modern architectures were able to make class-balanced predictions, yet still struggled with the Fear class. Moreover, CNN-based architectures proved to generally satisfy the interpretability target of this project, while the ViT-based model was background-prone and highly dependent on the images.

4.4 Results of the Combined Dataset Experiments

As Table 4.5 suggests, a new, larger dataset from Section 3.2.3 improved overall performance for all six models compared to the experiments previously conducted in this project. Although modern architectures from Section 3.2.2 still noticeably surpass those from the original work by Chávez-Guerrero et al. in [4], ResNet50 was able to achieve results very close to theirs. MobileNetV2, however, despite improved test set accuracy, clearly underfitted the training set with the best training accuracy of 0.8789, mainly due to the cosine annealing introduced for learning rate scheduling. While no validation set overfitting was detected, the held-out test set was relatively easier for some models.

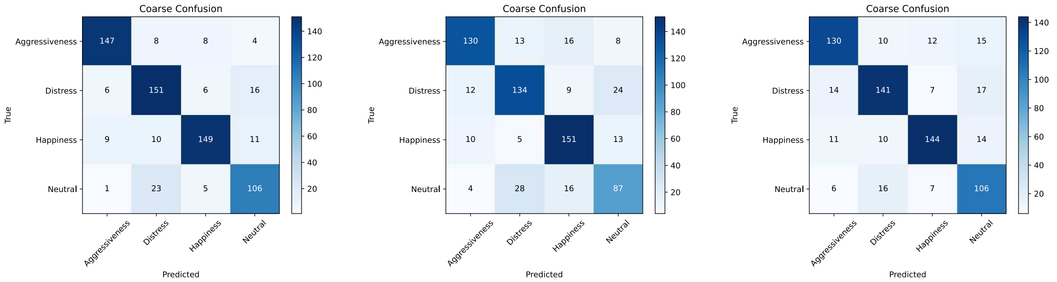
As far as the per-class performance is concerned, F1 scores were even across classes and models, with MobileNetV2 and VGG-16 relatively struggling with the most underrepresented Neutral class, as described in Section 3.2.3. Moreover, despite having the same support, all the models adapted to the Happiness class more effectively than to the Distress class. Per-class analysis further confirmed that EfficientNetV2-M, ConvNeXt-Base and DINOv2 ViT-S/14, with ResNet50 just behind them, achieved mostly similar results for all 4 classes of the new dataset.

Model	Test Acc	Macro F1	Weighted F1	Best Val	Worst Val	Best Train
ResNet50	0.8379	0.8358	0.8385	0.8215	0.7492	0.9983
VGG-16	0.7606	0.7543	0.7602	0.7803	0.5707	0.9924
MobileNetV2	0.7894	0.7872	0.7903	0.7685	0.4436	0.8789
EfficientNetV2-M	0.8545	0.8525	0.8561	0.8325	0.5210	0.9945
ConvNeXt-Base	0.8591	0.8574	0.8600	**0.8620**	0.5328	0.9996
DINOv2 ViT-S/14	**0.8621**	**0.8586**	**0.8619**	0.8552	0.4192	0.9983

Model	Class	Prec	Recall	F1	Support	Acc
ResNet50	Aggressiveness	0.9018	0.8802	0.8909	167	0.8802
	Distress	0.7865	0.8436	0.8140	179	0.8436
	Happiness	0.8869	0.8324	0.8588	179	0.8324
	Neutral	0.7737	0.7852	0.7794	135	0.7852
VGG-16	Aggressiveness	0.8333	0.7784	0.8050	167	0.7784
	Distress	0.7444	0.7486	0.7465	179	0.7486
	Happiness	0.7865	0.8436	0.8140	179	0.8436
	Neutral	0.6591	0.6444	0.6517	135	0.6444
MobileNetV2	Aggressiveness	0.8075	0.7784	0.7927	167	0.7784
	Distress	0.7966	0.7877	0.7921	179	0.7877
	Happiness	0.8471	0.8045	0.8252	179	0.8045
	Neutral	0.6974	0.7852	0.7387	135	0.7852
EfficientNetV2-M	Aggressiveness	0.9408	0.8563	**0.8966**	167	0.8563
	Distress	0.7884	0.8324	0.8098	179	0.8324
	Happiness	0.9310	0.9050	**0.9178**	179	0.9050
	Neutral	0.7586	0.8148	0.7857	135	0.8148
ConvNeXt-Base	Aggressiveness	0.9467	0.8503	0.8959	167	0.8503
	Distress	0.7908	0.8659	0.8267	179	0.8659
	Happiness	0.8994	0.8994	0.8994	179	0.8994
	Neutral	0.8074	0.8074	**0.8074**	135	0.8074
DINOv2 ViT-S/14	Aggressiveness	0.9177	0.8683	0.8923	167	0.8683
	Distress	0.8144	0.8827	**0.8472**	179	0.8827
	Happiness	0.8852	0.9050	0.8950	179	0.9050
	Neutral	0.8320	0.7704	0.8000	135	0.7704

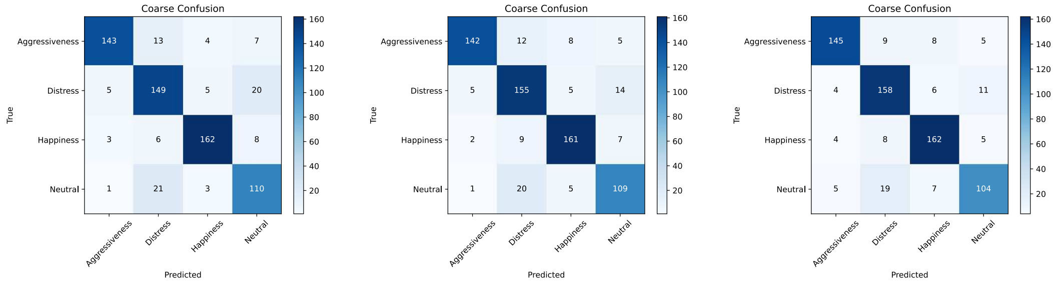
Table 4.5: Combined dataset experiments results.

Confusion matrices in Figure 4.18 also support everything said above regarding the model performances, providing more detailed insights into the common misclassification patterns. Clearly, the most common confusion for all models, and especially VGG-16, occurred between Neutral and Distress classes.



(a) ResNet50 (b) VGG-16 (c) MobileNetV2

Figure 4.18: Confusion matrices for the combined dataset experiments.



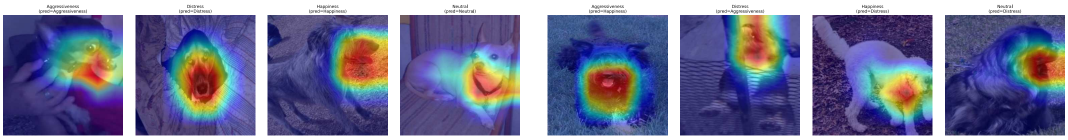
(d) EfficientNetV2-M

(e) ConvNeXt-Base

(f) DINOv2 ViT-S/14

Figure 4.18: Confusion matrices for the combined dataset experiments (continued).

ResNet50 EigenCAM heatmaps, shown in Figure 4.19, were very similar to the ones from Section 4.1 and Section 4.2, with the focus mainly on the chest and face. Although background pull was still present, the hotspot blobs became tighter.

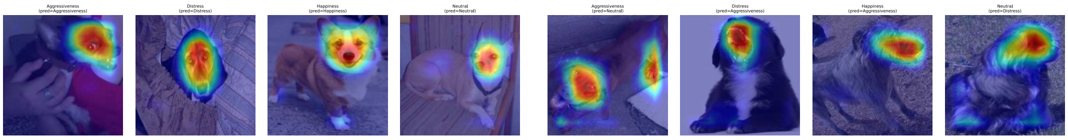

Aggressiveness (pred=Aggressiveness)
Distress (pred=Distress)
Happiness (pred=Happiness)
Neutral (pred=Neutral)
Aggressiveness (pred=Happiness)
Distress (pred=Aggressiveness)
Happiness (pred=Distress)
Neutral (pred=Distress)

(a) Correct

(b) Incorrect

Figure 4.19: EigenCAM overlays — ResNet50 with the combined dataset.

Although more rarely, VGG-16, as is common for the architecture itself, generates multi-blob heatmaps with minimal background pull. Moreover, a slight attention shift to the legs can be noticed in Figure 4.20. Despite a seemingly logical focus on the dog instead of the background, the relatively lower results shown in Table 4.5 suggest that clean localisation does not guarantee correct reasoning and that fragmented attention without the integration of whole-body cues is insufficient for reliable class discrimination.

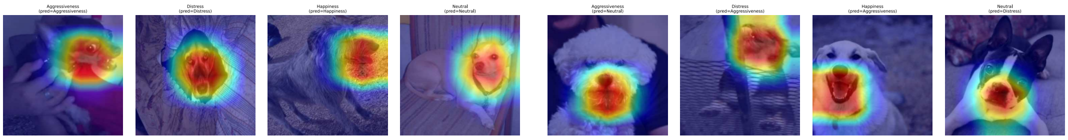

Aggressiveness (pred=Aggressiveness)
Distress (pred=Distress)
Happiness (pred=Happiness)
Neutral (pred=Neutral)
Aggressiveness (pred=Neutral)
Distress (pred=Aggressiveness)
Happiness (pred=Aggressiveness)
Neutral (pred=Distress)

(a) Correct

(b) Incorrect

Figure 4.20: EigenCAM overlays — VGG-16 with the combined dataset.

MobileNetV2 heatmaps were more stable than in the reproduction results and became more face-centric, as can be seen in Figure 4.21. Additionally, the background bleed was noticeably lower, aligning with the improved accuracy and F1 scores.

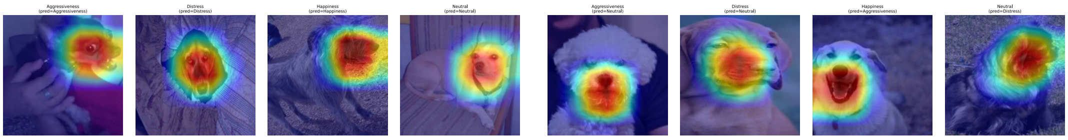

Aggressiveness (pred=Aggressiveness)
Distress (pred=Distress)
Happiness (pred=Happiness)
Neutral (pred=Neutral)
Aggressiveness (pred=Neutral)
Distress (pred=Aggressiveness)
Happiness (pred=Aggressiveness)
Neutral (pred=Distress)

(a) Correct

(b) Incorrect

Figure 4.21: EigenCAM overlays — MobileNetV2 with the combined dataset.

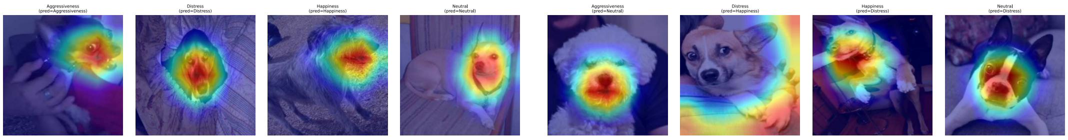
EfficientNetV2-M matches the model architecture experiment results in Section 4.3, generating compact head-focused heatmaps. Noticeably, as shown in Figure 4.22, the heatmaps were focused on the mouth for the Happiness class, matching the top distinctly high performance of the model shown in Table 4.5. High F1 scores and few background-prone misclassifications imply consistent capture of class-relevant cues.


Aggressiveness (pred=Aggressiveness)
Distress (pred=Distress)
Happiness (pred=Happiness)
Neutral (pred=Neutral)
Aggressiveness (pred=Neutral)
Distress (pred=Neutral)
Happiness (pred=Aggressiveness)
Neutral (pred=Distress)

(a) Correct (b) Incorrect

Figure 4.22: EigenCAM overlays — EfficientNetV2-M with the combined dataset.

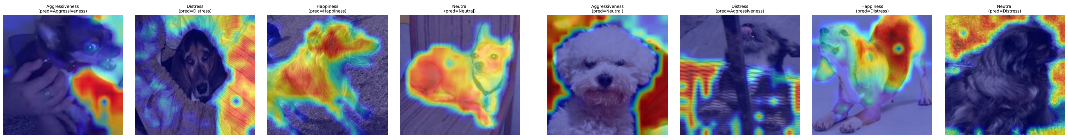
As Figure 4.23 shows, ConvNeXt-Base with the combined dataset produces EigenCAM overlays with good localisation of heatmaps and mostly face-centred focus, similarly to EfficientNetV2-M. Although in some cases it shows the strongest background attraction among the CNNs, overall, ConvNeXt-Base EigenCAM heatmap patterns are sufficiently reliable.


Aggressiveness (pred=Aggressiveness)
Distress (pred=Distress)
Happiness (pred=Happiness)
Neutral (pred=Neutral)
Aggressiveness (pred=Neutral)
Distress (pred=Happiness)
Happiness (pred=Distress)
Neutral (pred=Distress)

(a) Correct (b) Incorrect

Figure 4.23: EigenCAM overlays — ConvNeXt-Base with the combined dataset.

DINOv2 ViT-S/14 also exhibits a behaviour similar to that observed in the model architecture experiment results, generating the broadest, scene-level heatmaps with frequent background dominance. As can be observed in Figure 4.24, the combined dataset predictions were still based on the attention to either almost the whole body or entirely to the background. Such behaviour suggests that the high accuracy and F1 score of DINOv2 ViT-S/14 come from context memorisation rather than the recognition of emotion-specific features.


Aggressiveness (pred=Aggressiveness)
Distress (pred=Distress)
Happiness (pred=Happiness)
Neutral (pred=Neutral)
Aggressiveness (pred=Neutral)
Distress (pred=Aggressiveness)
Happiness (pred=Distress)
Neutral (pred=Distress)

(a) Correct (b) Incorrect

Figure 4.24: EigenCAM overlays — DINOv2 ViT-S/14 with the combined dataset.

The results of the combined dataset experiments further support those from Section 4.2 and Section 4.3 and show how the performance can be improved with a larger and more class-balanced dataset. Although the new dataset did not contain any significantly problematic classes, such as the Fear class in the original dataset, the models still somewhat struggled with the new Neutral class. EigenCAM heatmaps also demonstrate the interpretability advantages of CNNs compared to the ViT model, backing up the claims made in Section 4.3. Overall, considering both the performance and explainability of predictions, EfficientNetV2 and ConvNeXt performed the best out of the six presented models.

Chapter 5: Discussion

Although the project's methodology focuses mainly on the technical aspects of the implementation, its primary goal is to benefit those at the centre of the work — the dogs. Considering animal welfare and agency, the project does not aim to achieve the highest possible classification accuracy, but instead seeks a clear and understandable reasoning behind the model's predictions. A prediction matters only if it is grounded in the class-related cues, which is why so much attention was given to the EigenCAMs in Chapter 4, treating explainability as part of a validity check rather than a simple cosmetic add-on.

Labeling inconsistency and human bias are two other extremely important factors that shape predictions just as much as the model's capability to capture correct features. Especially, the difference in the reliability of the labeling is noticeable in the datasets where purely images and images extracted from the videos are used. For instance, in the original dataset from [4], the authors collected videos with situational context to extract the frames and label them based on that context. In contrast, the DEBiW dataset from [8] was created from internet images with no contextual information. Moreover, each of those images was labeled by multiple annotators with different judgments of the situation. In addition, many studies do not consider the unavoidable human bias that leads to labeling, dependent on the ethnic or cultural background of the labelers, which is not uncommon in machine learning. Of course, one can argue that these are not significant enough to bother spending time and resources on them; however, the reality is that without considering all the necessary factors, one cannot be certain of the reliability of the labels.

A related point is the common treatment of the dog's emotions as a set of discrete states, similarly to what has been implemented in this project. Despite resorting to such a coarse approach, it is necessary to acknowledge the flaws of being unable to cover the complete spectrum. One of those flaws, extremely important to classification with computer vision, is strangely yet tightly related to machine learning. Forcing the boundaries between emotions can lead to the model separating scenes instead of the actual emotions if the background patterns in the images correlate with one or the other side of a boundary. As Chapter 4 showed for DINOv2 ViT-S/14, high accuracy does not prove whether the predictions are valid or are the consequences of forced class boundaries. Obviously, from an ethical and logical perspective, it is also wrong to view emotions discretely, considering the original goal of emotion classification - animal welfare. For instance, if the intention is to detect early signs of a degenerative disease such as cognitive dysfunction syndrome, crisp boundaries lead to brittle outputs that hide, sometimes crucial, drifts in behaviour [41]. As a result, gradual loss of engagement or rising night-time distress patterns can be mistaken for noise, thus reducing the main aim to mere emotion labeling. Moreover, many datasets treat the emotions independently of the dogs' breed or age. When supervision does not account for factors such as breed-specific morphology and changes related to age, models can both learn shortcuts tied to breed or age and base their predictions on false markers. For example, drooping ears or senior stiffness can mislead the models towards distress or anxiety, when in reality, those may not be actual emotional cues.

Another discussion point, directly following the non-spectrum approach, is the underutilisation of the valence-arousal or any similar model in emotion-related studies. For example, despite collecting the valence-arousal data in [8], it is not utilised for further classification,

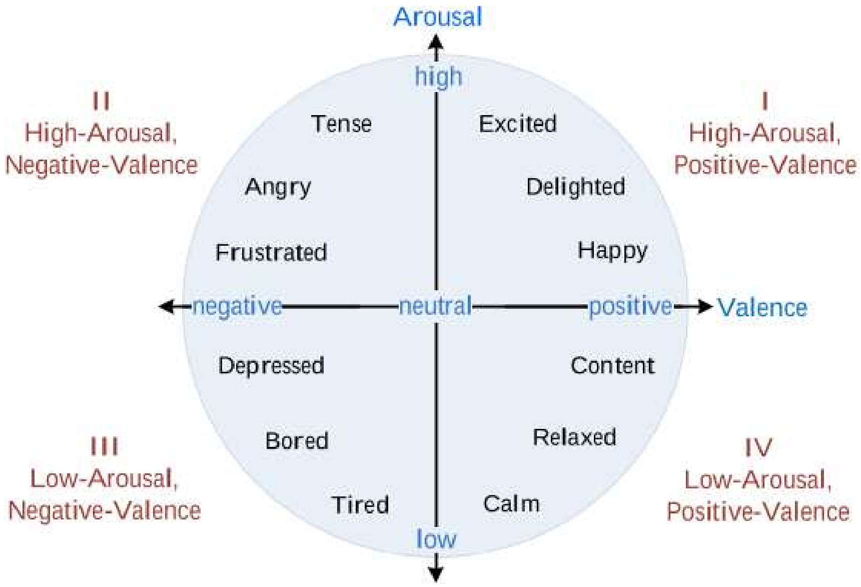


Figure 5.1: Circumplex model of emotions [42].

whereas it could potentially benefit emotion comprehension by the models. The circumplex model of emotions, shown in Figure 5.1, is one way to view emotions as a spectrum and gain more insights into the dog's behaviour, which might help humans analyse and models make more reliable predictions.

From a technical point of view, despite the small gains in accuracy, data augmentations taught the models variations of pose, lighting, viewpoint and coat to minimise the focus on these factors. In exploring model architectures, the main goal was to understand which inductive biases are prioritised by different networks. When models agree on dog-centred features, confidence in the findings increases. Conversely, if they diverge and attention shifts to the background, caution is required regarding the results, regardless of performance metrics. Furthermore, the significance of the contrast between CNN and ViT lies not in the superior performance of one over the other, but rather in the reasons behind such performance, which determines the safety of the outputs. As the project's results demonstrate, a larger dataset can provide broader visual coverage; however, it is merely another coarse approach to mitigate label drift and slightly weaken misleading associations.

Chapter 6: Limitations and Future Work

While the project's methodology ensures consistency in training and evaluation, adhering to common transfer learning guidelines, the extent to which its findings can be trusted is limited by several factors and design choices made primarily due to time constraints.

Firstly, splitting the data randomly, especially for the original dataset containing multiple images of the same dog in the same background, leads to the risk of information leakage from the training set to the validation and test sets. For future work, it is recommended to use grouped splits by clip, dog, or scene, essentially creating two smaller datasets and conducting cross-dataset training and evaluation to measure the domain shift. Moreover, to address the labeling issues mentioned in Chapter 5, expert ethology labels can be paired with external proxies such as thermal imaging or acoustic stress markers. Further developing the idea, the problem can potentially be addressed through full multi-modality by recording video and audio, aligning them in time, and utilising multi-modal classification models on short clips instead of single frames.

Secondly, following the discussion on discrete class boundaries from Chapter 5, soft labels can be used to maintain ambiguity and add valence and arousal-based subclasses. Alternatively, the task can be viewed as a regression problem, predicting continuous scores on valence and arousal scales, rather than a classification problem. Additionally, breed, age, and morphology should be taken into account when collecting the dataset, either by using only certain representatives of each group or by ensuring maximum diversity across all groups. To avoid the background bias, images with the background subtracted, similar to what is described in [4], can be used as input for the models as a coarse solution, or the images can be cropped to minimise scene-level information as a more delicate one.

Thirdly, the project limits the EigenCAM interpretation of predictions to informal inspection on a portion of images. It allows for understanding the general behaviour of the models; however, a better approach would be to report quantitative localisation against dog-part references using metrics such as Intersection-over-Union (IoU). In addition, for a better tracking of where any transformer-style model looks, using attention rollout would prevail in comparison to EigenCAM, which provides simple sanity checks. However, if the primary goal is to argue why a prediction points to a particular class, switching from EigenCAMs to Grad-CAM++ [43] is necessary, as it is computed from the gradients of the chosen class score with respect to feature maps. Finally, following the previous recommendations, if short clips are used instead of single frames, temporal consistency should also be tested by evaluating sliding windows and reporting the saliency overlap after optical flow warping [44].

Fourthly, one of the most significant issues in emotion classification with computer vision on images, and the sole reason for combining the three datasets in Section 3.2.3, is the amount and quality of the data. Considering that emotion, in its actual sense, is a broad concept, neural networks might experience difficulties in capturing the necessary features. Hence, if the plan is to use single frames, those frames should include information needed to guide the networks towards learning features crucial to emotion classification. In addition to the background-related solution proposed previously in this chapter, an obvious way to improve the data is to collect more data, maintaining the diversity. Alternatively, instead of using full images of the dogs, crops containing only specific emotion-based cues could help with attention shifts, or prediction-based data augmentation could encourage the models to learn features by creating artificial obstacles using methods like SaliencyMix, Attentive CutMix, etc [45], [46]. Another aspect worth mentioning is the availability of data, specifically the lack of open-access datasets. Despite a fully open release often being impossible because of data ownership and privacy, building larger, well-labeled datasets would benefit the field with controlled access as a practical middle ground. By utilising shared hubs with public metadata, running hosted benchmarks similar to those offered by Kaggle, but on a larger scale, and providing limited downloads, the availability could be improved, and the community would make more efforts to enhance the area.

Finally, some limitations do not apply to this project only, but rather are common to the entire area of animal emotion recognition. As mentioned in Chapter 2, the lack of a unified framework is often overlooked as a progress-limiting factor. For instance, evaluation practice lacks a shared standard, with results reported from single seeds and the rare use of cross-dataset validation, a practice this project also falls short of. A minimum protocol for future work should require grouped split, ultimate-seed repeats, and class-balanced metrics. Importantly, many carefully trained modern architectures, as this project shows, can perform well; thus, the consistent and rigorous evaluation is more of a bottleneck than the model choice. Therefore, it should be prioritised, as this project does, and preferably include as many recommendations as possible from the ones presented in this chapter.

Chapter 7: Conclusion

Repeating the Animal-Computer Interaction principle that motivates the project, technical choices and gains only matter in that case if the animals' agency and welfare are prioritised. The methodology was designed not only to make decisions but also to support them by explaining the attention of the models and choosing a conservative approach when evidence is weak.

Methodologically, the work introduces a repeatable pipeline to assess heterogeneous models on the same footing and incorporates interpretability as an integral part of validity, rather than an afterthought. The four-class dataset from multiple sources provides a fairer comparison between sources, and the unified evaluation of overall and per-class performance associates localisation metrics with saliency checks. Together, these elements show where CNNs typically focus on face and body cues, and where the ViT might overlook essential background details. This helps turn raw accuracy into meaningful insights about how the models arrive at their decisions.

The broader value lies in protocol, including consistent splits, class-balanced reporting, and, preferably, quantitative explainability with localisation against dog-part references, rather than architecture. This can enhance comparability across studies, while spectrum-based targets, such as valence–arousal, offer a path away from brittle, discrete labels. Data practice should remain light-touch and documented, with controlled access and opt-out routes, so that reliability gains do not come at the expense of the animal.

Translating these principles into practice, neural networks should be treated as decision support; thus, each prediction should be coupled with an evidence pack, analysing the decisions in detail. Although class-wise, contextual, and characteristic error analysis can be routine, it can also reveal common failure modes that require further adjustments through a simple improvement loop. Looking forward, practitioner-in-the-loop with as much expert analysis as possible, modest multi-modality with video, audio and thermal information, and quantified explainability for reliable predictions are the clearest paths to impact.

In conclusion, this project offers a clear path from computer vision methods to humane and accountable practice by keeping the dog's agency and welfare at the centre of every decision. It shows that modern models can be effective when guided by careful evaluation and interpretable, conservative behaviour, and it positions the system as support rather than replacement for human judgement. The work provides a coherent foundation for future refinement and collaboration, aiming to transform technical progress into dependable assistance in real-world contexts while remaining respectful of the animal.

Appendix A: Resources

A.1 Datasets

Original baseline dataset

Access can be requested from the authors of the replicated study using an institutional email. The publication page can serve as a starting point for contact and citation: doi:10.13053/cys-26-1-4165.

DEBiW dataset

DEBiW was recommended by Dr Humberto Pérez-Espinosa during this project. Access can be obtained by contacting Dr Pérez-Espinosa or the other dataset authors; the formal reference can be consulted here: doi:10.1145/3565995.3566041.

Kaggle Dog Emotion dataset

The Kaggle Dog Emotion dataset can be accessed openly at doi:10.34740/KAGGLE/DSV/4969612. Download and use can follow Kaggle's terms.

A.2 Code and outputs

All scripts, logs, curves, confusion matrices, and saliency overlays are available at github.com. Accessed: 1 Sept. 2025. The README describes setup and reproduction steps that can be followed to rerun the experiments.

A.3 Related work published during the project

A multimodal study on canine emotion classification by Dr. Pérez-Espinosa et al., somewhat following the recommendations of this project regarding multimodality, was published while this project was in progress. The paper can be accessed here: doi:10.1016/j.patrec.2025.06.018.

Bibliography

[1] Clara Mancini. Animal-computer interaction: a manifesto. *Interactions*, 18(4):69–73, July 2011. doi:10.1145/1978822.1978836.

[2] Ilyena Hirskyj-Douglas, Patricia Pons, Janet C Read, and Javier Jaen. Seven years after the manifesto: Literature review and research directions for technologies in animal computer interaction. *Multimodal Technologies and Interaction*, 2(2):30, 2018. doi:10.3390/mti2020030.

[3] Rebecca Kleinberger, Lena Ashooh, Keavan Farsad, and Ilyena Hirskyj-Douglas. Animals' entanglement with technology: a scoping review. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–22, 2025. doi:10.1145/3706598.3713384.

[4] Víctor Ocyel Chavez-Guerrero, Humberto Perez-Espinosa, María Eugenia Puga-Nathal, and Veronica Reyes-Meza. Classification of domestic dogs emotional behavior using computer vision. *Computación y Sistemas*, 26(1):203–219, 2022. doi:10.13053/cys-26-1-4165.

[5] B. Waller, Catia Correia Caeiro, K. Peirce, A. Burrows, and J. Kaminski. DogFACS: the dog facial action coding system. 2013. Accessed: 1 Sept. 2025. URL: https://repository.lincoln.ac.uk/articles/book/DogFACS_the_dog_facial_action_coding_system/24341533.

[6] Ádám Miklósi and József Topál. What does it take to become 'best friends'? evolutionary changes in canine social competence. *Trends in cognitive sciences*, 17(6):287–294, 2013. doi:10.1016/j.tics.2013.04.005.

[7] Gabriella Tami and Anne Gallagher. Description of the behaviour of domestic dog (canis familiaris) by experienced and inexperienced people. *Applied Animal Behaviour Science*, 120(3-4):159–169, 2009. doi:10.1016/j.applanim.2009.06.009.

[8] Fernanda Hernández-Luquin, Hugo Jair Escalante, Luis Villaseñor-Pineda, Verónica Reyes-Meza, Luis Villaseñor-Pineda, Humberto Pérez-Espinosa, Verónica Reyes-Meza, Hugo Jair Escalante, and Benjamin Gutierrez-Serafín. Dog emotion recognition from images in the wild: Debiw dataset and first results. In *Proceedings of the ninth international conference on animal-computer interaction*, pages 1–13, 2022. doi:10.1145/3565995.3566041.

[9] Tali Boneh-Shitrit, Marcelo Feighelstein, Annika Bremhorst, Shir Amir, Tomer Distelfeld, Yaniv Dassa, Sharon Yaroshetsky, Stefanie Riemer, Ilan Shimshoni, Daniel S Mills, et al. Explainable automated recognition of emotional states from canine facial expressions: the case of positive anticipation and frustration. *Scientific reports*, 12(1):22611, 2022. doi:10.1038/s41598-022-27079-w.

[10] Alberto C Villaluz, Joel C De Goma, Jianina Vennice T Besa, Jericho Ivan D Ignacio, and Stephanie Anne A Zaguirre. Emotion classification in domestic dogs using computer vision based on the dog's body and face. In *Proceedings of the 2024 9th International Conference on Intelligent Information Technology*, pages 359–364, 2024. doi:10.1145/3654522.3654558.

[11] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019. doi:10.1186/s40537-019-0197-0.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. doi:10.1109/cvpr.2016.90.

[13] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. doi:10.48550/arXiv.1409.1556.

[14] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. doi:10.48550/arXiv.1704.04861.

[15] Michelle Carney, Barron Webster, Irene Alvarado, Kyle Phillips, Noura Howell, Jordan Griffith, Jonas Jongejan, Amit Pitaru, and Alexander Chen. Teachable machine: Approachable web-based tool for exploring machine learning classification. In *Extended abstracts of the 2020 CHI conference on human factors in computing systems*, pages 1–8, 2020. doi:10.1145/3334480.3382839.

[16] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. doi:10.1109/cvpr.2018.00474.

[17] Suvaditya Mukherjee. The annotated resnet-50. towardsdatascience.com, 2022. Accessed: 1 Sept. 2025.

[18] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. doi:10.48550/arXiv.1412.6980.

[19] Rohini G. Everything you need to know about vgg16. medium.com, 2021. Accessed: 1 Sept. 2025.

[20] Zdzisław Kowalczuk, Michał Czubenko, and Weronika Żmuda Trzebiatowska. Categorization of emotions in dog behavior based on the deep neural network. *Computational Intelligence*, 38(6):2116–2133, 2022. doi:10.1111/coin.12559.

[21] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018. doi:10.48550/arXiv.1803.08375.

[22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. doi:10.9734/ajrcos/2025/v18i2569.

[23] Ulzhalgas Seidaliyeva, Daryn Akhmetov, Lyazzat Ilipbayeva, and Eric T Matson. Real-time and accurate drone detection in a video with a static background. *Sensors*, 20(14):3856, 2020. doi:10.3390/s20143856.

[24] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. doi:10.1109/iccv.2019.00612.

[25] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. doi:10.48550/arXiv.1710.09412.

[26] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020. doi:10.1609/aaai.v34i07.7000.

[27] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020. doi:10.48550/arXiv.2004.10934.

[28] Wei-Chao Cheng, Tan-Ha Mai, and Hsuan-Tien Lin. From smote to mixup for deep imbalanced classification. In *International Conference on Technologies and Applications of Artificial Intelligence*, pages 75–96. Springer, 2023. doi:10.1007/978-981-97-1711-8_6.

[29] Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pages 10096–10106. PMLR, 2021. doi:10.48550/arXiv.2104.00298.

[30] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*, 2016. doi:10.48550/arXiv.1605.07648.

[31] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019. doi:10.48550/arXiv.1906.02629.

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. doi:10.48550/arXiv.1711.05101.

[33] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. doi:10.48550/arXiv.1608.03983.

[34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. doi:10.1109/cvpr52688.2022.01167.

[35] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023. doi:10.48550/arXiv.2304.07193.

[36] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. doi:10.48550/arXiv.1607.06450.

[37] Daniel Shan Balico. Dog emotion, 2023. doi:10.34740/KAGGLE/DSV/4969612.

[38] Rahima Khanam and Muhammad Hussain. What is yolov5: A deep look into the internal features of the popular object detector. *arXiv preprint arXiv:2407.20892*, 2024. doi:10.48550/arXiv.2407.20892.

[39] Marina Sokolova and Guy Lapalme. A systematic analysis of performance measures for classification tasks. *Information processing & management*, 45(4):427–437, 2009. doi:10.1016/j.ipm.2009.03.002.

[40] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2020. doi:10.1109/ijcnn48605.2020.9206626.

[41] Makiko Ozawa, Mai Inoue, Kazuyuki Uchida, James K Chambers, Yukari Takeuch, and Hiroyuki Nakayama. Physical signs of canine cognitive dysfunction. *Journal of Veterinary Medical Science*, 81(12):1829–1834, 2019. doi:10.1292/jvms.19-0458.

[42] Tori Acres. Canine arousal: Optimal training zone vs over arousal. caninebodybalance.com.au, 2022. Accessed: 1 Sept. 2025.

[43] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)*, pages 839–847. IEEE, 2018. doi:10.1109/wacv.2018.00097.

[44] Wei-Sheng Lai, Jia-Bin Huang, Oliver Wang, Eli Shechtman, Ersin Yumer, and Ming-Hsuan Yang. Learning blind video temporal consistency. In *Proceedings of the European conference on computer vision (ECCV)*, pages 170–185, 2018. doi:10.1007/978-3-030-01267-0_11.

[45] A.F.M. Uddin, Mst. Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae, et al. Saliencymix: A saliency guided data augmentation strategy for better regularization. *arXiv preprint arXiv:2006.01791*, 2020. doi:10.48550/arXiv.2006.01791.

[46] Devesh Walawalkar, Zhiqiang Shen, Zechun Liu, and Marios Savvides. Attentive cutmix: An enhanced data augmentation approach for deep learning based image classification. *arXiv preprint arXiv:2003.13048*, 2020. doi:10.1109/icassp40776.2020.9053994.