VIBA-Net: A Multimodal Framework for Infant Behavior Annotation

William Ong
University of Washington
wong2@uw.edu

Sam Shin
University of Washington
seunguk@uw.edu

Raghav Ramesh
University of Washington
raghavtr@uw.edu

Abstract

Infant development research relies on careful observation of infant interactions with caregivers, toys, and auditory stimuli. We propose VIBA-Net, a privacy-preserving and computationally efficient model designed to analyze video recordings of infant interactions. By combining facial expression analysis, object detection, and audio analysis, VIBA-Net efficiently classifies emotional states, behavioral responses, and other key infant developmental indicators. Trained and evaluated on a curated dataset of annotated infant interactions, our model automates the annotation process and provides valuable insights into early infant development. VIBA-Net provides a comprehensive approach to better understanding infant behavior, with potential applications in both research and developmental monitoring.

1. Introduction

The UW Institute for Learning and Brain Sciences (I-LABS) analyzes videos of parent-infant interactions to better understand early human development. These observations help describe how infants process music and speech, how their experiences influence learning, and how early development impacts future outcomes. The videos include both auditory and non-auditory stimuli, such as physical and verbal interactions with parents or surroundings, which elicit various infant responses and emotional changes.

Due to HIPAA regulations and privacy concerns, existing high-accuracy LLMs cannot be adopted for such annotation tasks. As a result, researchers must manually annotate videos, a labor-intensive and time-consuming process that often detracts from core scientific work.

To streamline this workflow, we propose automating the annotation process. Our goal is to develop a lightweight model that preserves high accuracy while being efficient for real-time use—potentially through a mobile or desktop application, where smaller models are preferred for ease of deployment. The model will focus on extracting key features related to the infant(s) in each video, including emotional expressions, relevant objects, and audio cues.

2. Related Work

State-of-the-art multimodal large language models (LLMs) are very effective at annotating variable-length video input. However, their closed-source nature and data retention policies conflict with HIPAA regulations. While open-source models like Video-LLaMA and MiniGPT4-Video avoid these issues, they demand significant computational resources, making the annotation process costly and highly inefficient.

In the audio modality, there are three primary classification approaches: thresholding, rule-based classification, and machine learning-based classification. Each method often involves spectral analysis, where audio signals are visually represented as spectrograms that highlight the unique patterns of audio events. Common spectral analysis techniques include the Fourier Transform, which converts a time-domain signal into a frequency domain, and Mel-Frequency Cepstral Coefficients (MFCCs), which capture the short-term power spectrum of sound [1]. Thresholding involves classification based on whether a specific feature, such as a spectral centroid, exceeds a predefined threshold value [6]. Rule-based classification models are similar, but combine multiple features (e.g., MFCCs, spectral centroid) through a series of rules [8]. However, these methods are only effective when the features are distinguishable and belong to simple, single-class events. In contrast, the machine learning-based approach is significantly more flexible and robust. With various implementations and architectures, it can classify data featuring multiple audio classes or hard-to-distinguish features. Since infant interactions often have overlapping or varied audio events, we will follow the machine learning approach.

AudioSet is a key dataset for developing machine learning models for audio classification. The dataset includes over 2 million labeled video clips, along with 128-dimensional audio features extracted at 1Hz [2]. These features capture the high-level characteristics of each audio segment, such as timbre, pitch, and rhythm, which help distinguish different sound events. The features were extracted with Google VGGish, a CNN-based model inspired by VGG and trained on a preliminary version of YouTube-

8M [4]. To optimize efficiency and scalability, we will leverage these pre-trained features to build our classification model for sound events related to infant interactions.

In the visual modality, emotion recognition from images is a challenging task due to the subtlety and variability of facial expressions, as well as external factors like lighting and occlusion. While popular Convolutional Neural Networks (CNNs) like ResNet have demonstrated strong performance by learning hierarchical and abstract features directly from data, their large size and computational demands make them impractical for our real-time applications [3]. This leads to MobileNetV2, which offers a compelling balance between performance and efficiency [7]. Its use of inverted residual blocks and linear bottlenecks facilitates efficient feature reuse and gradient flow, while depthwise separable convolutions significantly reduce computation and memory requirements. These design choices are well-suited for real-time emotion recognition, especially in settings where efficiency is important, such as at I-LABS. However, despite being pre-trained on diverse datasets, MobileNetV2's lightweight architecture limits its ability to capture the fine-grained emotional cues needed for high-accuracy emotion recognition. As a result, we will explore how architectural modifications and parameter tuning can improve MobileNetV2's accuracy for emotion recognition in infants.

3. Methods

Designed to efficiently analyze audiovisual content and extract meaningful annotations, our model will process each input video through a three-stage pipeline: (1) fixed-rate sampling, (2) parallel video and audio feature analysis, and (3) feature aggregation for the final annotation.

The visual processing stream begins with uniform frame sampling at 1 fps. We adopt this rate to reduce the overall frame count, thereby lowering computational demands while retaining key visual cues.

We fine-tuned MobileNetV2 to predict babies' emotions from individual frames. Since MobileNetV2 was originally trained on general images rather than baby faces, the baseline model did not perform well, with an accuracy worse than random guessing. In order to make the most use of our limited dataset, we utilized an MTCNN to extract up to M facial croppings from images. After this is done for each image, they are batched for training. Additionally, we replaced the original output layer with a classification layer that categorizes emotions into three emotion classes: calm, laughing, and crying. However, due to the limited size and diversity of our dataset, the model overfit to the training data and demonstrated poor generalization. To mitigate this, we introduced a dropout layer. After tuning the dropout probability, a rate of 0.3 was found to be optimal. Further analysis revealed that the primary issue was class imbalance, with the dataset containing significantly more calm samples compared to laughing or crying samples. To address this, we augmented the crying and laughing classes using techniques such as horizontal flipping, color jittering, and slight rotations. We also experimented with the intensity of the data augmentation. Excessive augmentation led to slow convergence and poor training and validation loss, while insufficient augmentation failed to reduce validation loss. The optimal configuration was found to be horizontal flips with a probability of 0.5, rotations of 5 degrees, and mild color jittering with brightness, contrast, saturation, and hue. It was found that a learning rate of $1e^{-4}$ and a batch size of 32 was optimal for convergence . Additionally, we attempted using class weights in the cross-entropy loss function to address this imbalance. However, this approach proved to be less effective than data augmentation, which had yielded better performance improvements.

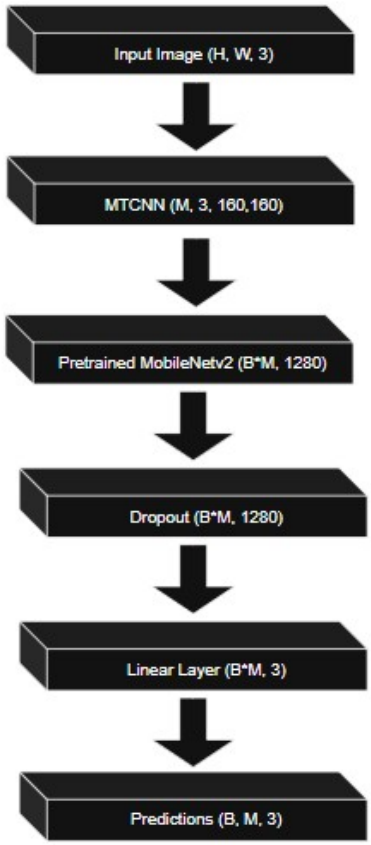

Figure 1. Fine-Tuned MobileNetv2 Architecture

We fine-tuned YOLOv8n, the smallest model in the YOLOv8 family, to efficiently detect people, estimate their age category, and identify objects within each video frame. To create the ground truth for object detection and bounding

box localization, we leveraged AWS Rekognition, a state-of-the-art cloud based computer vision service. Rekognition provided high confidence detections and bounding box coordinates for each object of interest, which were then reviewed and used to label our custom dataset. For training, we adopted the YOLOv8n pretrained weights as a starting point, prioritizing computational efficiency and faster convergence. The dataset was formatted according to the YOLO annotation schema, and training was configured with the following hyperparameters: 100 epochs, an image size of 640 pixels, batch size of 80, and the AdamW optimizer with an initial learning rate of 0.01. To prevent overfitting and enable early stopping, we set the patience parameter to 20 epochs and performed training with two data loading workers. All training was performed using the Ultralytics YOLO framework on NVIDIA T4 16GB VRAM GPU. Data augmentation techniques followed the YOLOv8 default settings, including random horizontal flips, scaling, and color jittering to enhance generalization and robustness. The final fine-tuned model was evaluated on the held-out validation and test splits, with precision, recall, and mean average precision (mAP) calculated at an IoU threshold of 0.5. This lightweight YOLOv8n configuration enabled rapid experimentation and inference, making it well-suited for integration into our real-time VIBA-Net annotation pipeline.

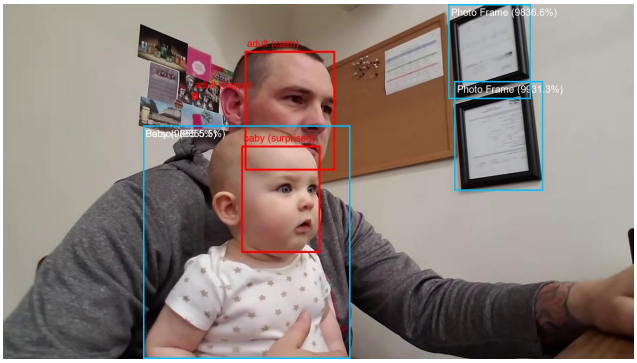

Figure 2. Example ground truth image frame

While fine-tuning YOLOv8n enabled the model to detect new object classes that were not present in the original pretrained weights, the results reveal significant limitations imposed by the small size of our training dataset. As shown in the figure below, the model's validation losses remain high and unstable, and both precision and mAP metrics are notably low across epochs. These patterns suggest that the model struggled to generalize, likely due to the limited diversity and quantity of labeled examples available for training. The lack of data is further reflected in erratic metric curves and potential overfitting, particularly for underrepresented classes. Nevertheless, the fine-tuned model successfully learned to recognize custom object categories specific to our annotation pipeline—categories that the original YOLOv8n model was unable to detect. This demonstrates the practical benefit of domain adaptation, but also highlights the need for a larger, more balanced training set to realize robust, real-world performance.

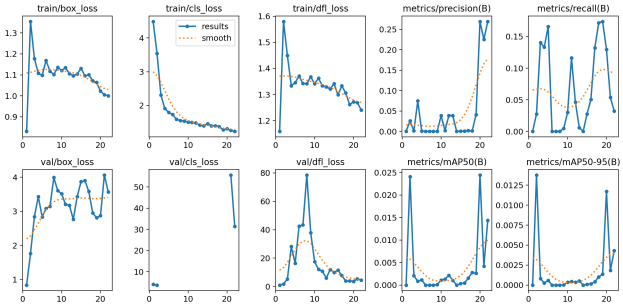

Figure 3. YOLOv8n Training Curves: Loss and mAP over Epochs

In parallel, the audio processing pipeline extracts the audio from the video, converts it to a mono track, and resamples it to a standard 16 kHz frequency. The audio is then segmented into 10-second intervals, with shorter segments being zero-padded for consistency. Each segment is transformed into a log-Mel spectrogram, from which the Google VGGish model extracts 128-dimensional features for classification. The VGGish features are classified into 8 different classes by our custom model, VGGishNet, whose architecture is shown in Fig. 4. The model includes a temporal average pooling layer to compress time-series data and reduce computational overhead, followed by two hidden layers incorporating batch normalization, LeakyReLU activation, and dropout regularization. The final output layer produces a binary encoding for the presence or absence of each sound class. The architecture, including layer sizes, number of hidden layers, activation functions, dropout rates, and other hyperparameters, was optimized through extensive tuning, using random search and accuracy curves as the primary methods for hyperparameter selection.

The visual and audio outputs are then aggregated by frame to generate the final output, which includes the number of people per-video, per-frame emotion labels, detected object interactions, and identified audio events. The overall model architecture is described in the following component flows and shown in Fig. 5.

- **Visual Stream:** Video Frames → Pre-processing → Fine-tuned MobileNetv2 → Emotion Annotations
  Video Frames → Fine-tuned YOLOv8n → Detected Objects Annotations
- **Audio Stream:** Audio Input → Pre-processing → Google VGGish → VGGishNet → Audio Classifications
- **Feature Aggregation:** Visual Annotations + Audio

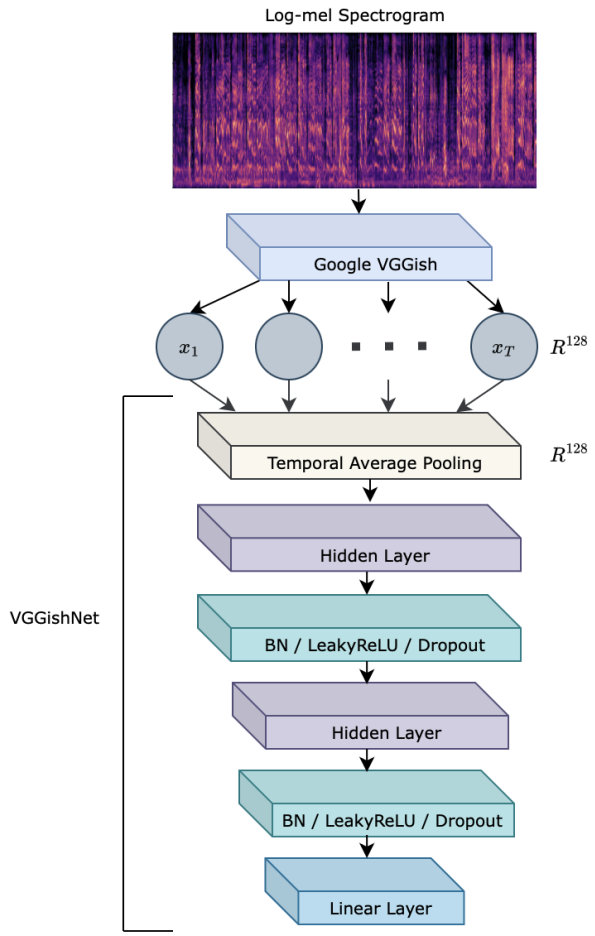

Figure 4. VGGishNet Architecture

Annotations → Fusion → Final Video Annotations

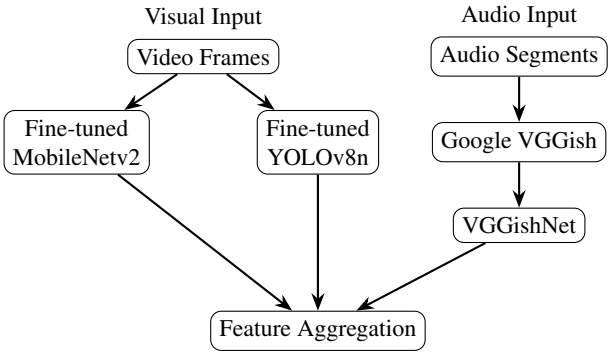

Figure 5. VIBA-Net Architecture

4. Experiments

4.1. Dataset

We collected and manually annotated 35 video clips of parent-infant interactions, ranging from 10 seconds to 3 minutes in length. To standardize our pipeline inputs, every clip is sampled at 1 frame per second, yielding between 10 to 180 frames per video (∼1,250 frames total). Annotations conform to our defined JSON schema and include:

- **Video metadata:** file name, duration (s), frame rate, and frame count
- **People summary:** total people, number of babies, and number of adults
- **Per-frame annotations:** for each frame, a list of detected people (unique IDs, type = {baby, adult}, emotion ∈ {calm, happy, mad, upset, intrigued}, interacted_objects) and the set of all objects present
- **Audio events:** for each continuous segment, start/end timestamps, event type (e.g. baby crying, music), and an assigned confidence score

We partitioned the 35 clips into a 70/20/10 train, validation, test split, with 24, 7, and 4 clips in the training, validation, and test sets, respectively. The test set is reserved for the final evaluation (Sec. 4.4), the validation set is used for hyperparameter tuning and early stoppage, and the train split is used for model training.

VGGishNet also used the extracted VGGish features from AudioSet, with approximately 300 training features and 50 test features for each of the 8 classes, where classes = {baby crying, baby laughing, music, singing, child speech, male speech, female speech, lullaby}.

4.2. VGGishNet Tuning and Architecture Selection

We used random search to optimize VGGishNet's architecture and hyperparameters, including learning rate, weight decay, batch size, dropout rate, hidden layer size, and number of hidden layers. We chose random search over grid search for its efficiency and flexibility in exploring a more complex hyperparameter space. Random search can achieve comparable or even better model performance with fewer evaluations compared to grid search [5]. Out of 40 random architectures with 100 epochs and early stopping of patience = 10, the best-performing configuration had 2 hidden layers with sizes 64 and 32, dropout rate = 0, weight decay = 0.001, learning rate = 0.0001, and batch size = 64. This configuration achieved 91% accuracy on the test set and 95% accuracy on the training set (Fig. 6). In addition, we experimented with different activation functions, in which the LeakyReLU activation outperformed standard ReLU, providing a 0.6% improvement in test accuracy and over 2.3% improvement in training accuracy. We also tried various models, including simple multi-layer perceptrons,

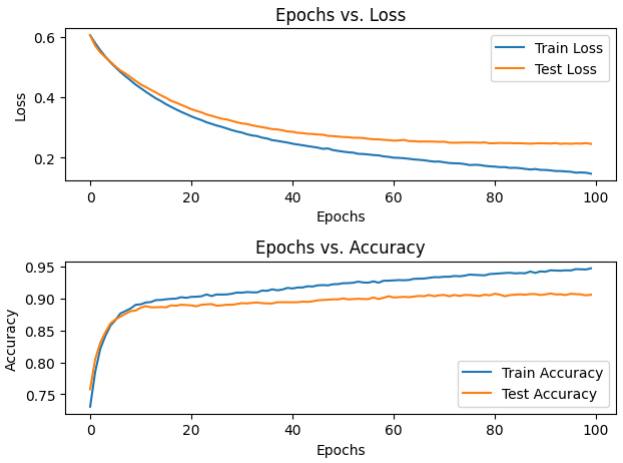

Figure 6. VGGishNet Training Plots

more complex CNNs with attention, and Bi-LSTMs with attention, but none of the models outperformed the 2-hidden-layer neural network. Lastly, we tested VGGishNet without the temporal average pooling layer and observed no significant changes in accuracy. As a result, we decided to retain it, as keeping the layer would reduce computational costs.

4.3. Hyperparameter Training

For VIBA-Net, we similarly used random search to optimize its performance. We used the validation set for all tuning decisions and applied early stopping based on validation loss to prevent overfitting. We tuned the learning rate, batch size, and dropout rate. Each configuration was evaluated using the following composite score:

$$\begin{aligned}\text{Score} = \lambda_1 \cdot \text{Emotion Accuracy} \\ + \lambda_2 \cdot \text{Audio Event F1} \\ + \lambda_3 \cdot \text{Mean IoU (Object Detection)}\end{aligned}$$

with equal weights $\lambda_i = 1$.

4.4. Evaluation Protocol

To quantitatively compare our model against baselines (e.g. CogVLM-17B, Gemini Pro), we measure annotation accuracy via field-level matching between predicted and ground-truth JSON:

Matching Rules

- **Counts** (`total_people`, `num_babies`, `num_adults`): exact integer match.
- **Emotions**: exact match within the emotion label set
- **Objects**: compare predicted and ground-truth object sets using semantic similarity. A match is accepted if the cosine similarity between their `text-embedding-3-large` (OpenAI) embeddings is ≥ 0.75.
- **Frame indices / timestamps**: exact match at 1 fps sampling
- **Audio events**: accept predictions with confidence ≥ 0.5. Match to a ground-truth event if (i) types coincide and (ii) temporal IoU ≥ 0.5

4.5. Results

We aim to perform better than the current state of the art LLMs in latency and compute/memory usage.

Model	Mem (GB)	Lat (ms)	Params (M)	Ppl F1 (%)	Obj F1 (%)	Cls Acc (%)
VIBA-Net	0.04	41	3.6	57.9	6.9	37.0
Qwen3B	12	41811	3000	60.9	10.5	84.6
Qwen7B	20	32886	7000	52.5	12.6	82.4
Qwen32B	80.0	75652	32000	61.2	10.4	52.5

Table 1. Detection and Classification Metrics.

Table 1 summarizes the detection and classification metrics across models, along with corresponding memory usage, latency, and parameter count. VIBA-Net achieves a competitive people detection F1-score (57.9%) while maintaining extremely low memory requirements and rapid inference speed (41 ms per frame), significantly outperforming large LLM-based models in computational efficiency. Although object detection and classification accuracy remain lower compared to the Qwen models—reflecting the limited size and diversity of the training data—VIBA-Net demonstrates the ability to detect custom classes specific to our application domain. These results highlight the trade-off between accuracy and efficiency, as well as the practicality of lightweight models for resource-constrained environments.

MobileNetv2:	Fine-Tuned	Baseline
Calm Accuracy	0.9	0.0
Crying Accuracy	0.3	0.1
Laughing Accuracy	0.6	0.1

Table 2. MobileNetv2 Performance

To ensure that the fine-tuned MobileNetv2 model generalized well, the model was tested on a set of random public images totaling to nearly 10 % of the original dataset. While the crying accuracy remained relatively low due to a lack of good data surrounding crying emotions in babies, the accuracy in accurately predicting calm and laughing went up significantly from the baseline pretrained MobileNetv2 model (Table 2).

5. Discussion

Given a video of parent-infant interactions, our model will classify and annotate the video at various time steps, including various stimuli (auditory and visual), along with infant responses. We intend to assist the researchers at I-LABS with our model, which will remain open-source to respect privacy concerns. In addition, it will be efficient and accurate, producing results that assist researchers in better understanding early human development.

5.1. Limitations

It is important to note that our model may not generalize well to all scenarios. Given the ethical considerations surrounding the use of baby videos, our ability to collect clean and relevant data was limited, resulting in a relatively small dataset of just 35 annotated videos. Using videos was an issue given that each frame in the video was highly correlated to other frames, posing an overfitting issue. Additionally, this constrained sample does not capture the full diversity of real-world parent–infant interactions. Different cultural contexts, lighting conditions, camera angles, or infant ages could all affect model performance. Another limitation lies in the annotations themselves. Although each video was labeled by a human, manual labeling is inherently time consuming and prone to occasional inconsistencies or oversights, particularly for subtle or ambiguous emotional expressions. Moreover, our pipeline currently treats audio and visual streams separately before aggregation, which may overlook fine grained audio–visual correlations. Integrated training could better align these modalities. Our fixed 1 fps frame sampling simplifies computation but may miss brief events suggesting that future work should explore adaptive or higher-frequency sampling strategies.

6. Conclusion

In this work, we present VIBA-Net, a lightweight, multimodal annotation framework tailored for analyzing infant-caregiver interactions. Our system integrates fine-tuned MobileNetv2 for facial emotion recognition, YOLOv8n for object and person detection, and VGGishNet for audio event classification. Despite the constraints of limited data and real-world variability, VIBA-Net demonstrates competitive performance in people detection while maintaining exceptional efficiency by more than a magnitude and privacy compliance. By automating video annotation through a modular pipeline, our framework offers a scalable tool to support infant development research and opens new possibilities for accessible behavioral analysis across diverse settings. Future work will focus on improving generalizability through larger datasets and integrated audio-visual training strategies, paving the way for a more practical deployment in clinical or research settings.

References

[1] Zrar Kh. Abdul and Abdulbasit K. Al-Talabani. Mel frequency cepstral coefficient and its applications: A review. *IEEE Access*, 10:122136–122158, 2022.

[2] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[4] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification, 2017.

[5] Kimberly Jane. Hyperparameter optimization: Fine-tuning model hyperparameters using techniques like grid search or random search. 10 2024.

[6] Thomas Pellegrini and Timothée Masquelier. Fast threshold optimization for multi-label audio tagging using surrogate gradient learning, 2021.

[7] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks, 2019.

[8] Jozef Vavrek, Jozef Juhár, and Anton Čižmár. Audio classification utilizing a rule-based approach and the support vector machine classifier. In *2013 36th International Conference on Telecommunications and Signal Processing (TSP)*, pages 512–516, 2013.