Emotion-Aware Music Recommendation

Hieu Tran[*]**, Tuan Le**[*]**, Anh Do**[*]**, Tram Vu, Steven Bogaerts, Brian Howard**
DePauw University
602 S. College Ave, Greencastle, IN 46135, U.S.A.
{hieutran_2023, tuanle_2024, phuonganhdo_2023, tramvu_2024, stevenbogaerts, bhoward}@depauw.edu

Abstract

It is common to listen to songs that match one's mood. Thus, an AI music recommendation system that is aware of the user's emotions is likely to provide a superior user experience to one that is unaware. In this paper, we present an emotion-aware music recommendation system. Multiple models are discussed and evaluated for affect identification from a live image of the user. We propose two models: DRViT, which applies dynamic routing to vision transformers, and InvNet50, which uses involution. All considered models are trained and evaluated on the AffectNet dataset. Each model outputs the user's estimated valence and arousal under the circumplex model of affect. These values are compared to the valence and arousal values for songs in a Spotify dataset, and the top-five closest-matching songs are presented to the user. Experimental results of the models and user testing are presented.

1 Introduction

It is clear that emotions influence behavior and preferences. Some AI systems are intentionally *human-aware*, considering characteristics including emotions to improve the user experience. Music recommendation is one area in which emotions should logically play a role, as people frequently listen to music that matches their current mood (Juslin, Sloboda et al. 2001). An AI system that is aware of emotions expressed through music can make recommendations to help users regulate their feelings or boost their current mood (Taruffi et al. 2017). More precisely, we distinguish here between *emotions*, a feeling inside that is not directly observable, and *affect*, the external expression of emotions, particularly in the face. In this study, we develop a music recommendation system aiming to identify affect from the user's face and recommend songs determined to fit that affect most closely. By estimating affect, the system aims to be emotion-aware.

This paper is organized as follows. Section 2 discusses prior work on music recommendation systems and affect identification. Section 3 describes the major components of our system. Section 4 contextualizes our affect identification models in neural network research, focusing on involution and dynamic routing for vision transformers. Section 5 describes our experimental setup for testing the affect identification models, with the experimental results and analysis in Section 6. Section 7 describes the process by which the identified affect is used to recommend songs, and the results of user experiments are presented in Section 8. Section 9 provides conclusions and future work for the research.

2 Related Work

Many AI techniques have been applied to music recommendation. Ji et al. (2015) propose a time-based Markov embedding to observe user music selections over time for further music recommendations. Logan (2004) uses an acoustic-based similarity measure to group related songs into "song sets" for recommendation. Hsu et al. (2016) propose a natural language processing model called CNN-rec to recommend music based on the user's recent listening history. Hu and Ogihara (2011) use an autoregressive integrated moving average (ARIMA) model to estimate the genre, year, and "freshness" of the user's song selections and thus learn the user's preferences. Samuvel, Perumal, and Elangovan (2020) extract an "EigenFace" vector expressing important portions of the user's face. This is fed to a support vector machine to estimate affect and recommend music. In another support vector machine approach, James et al. (2019) preprocess facial images into a sequence of "action units" used to classify emotion for music recommendation.

There is also much related work in affect identification. For example, convolution-based models are typical in image tasks, and can be applied to affect identification specifically (Giannopoulos, Perikos, and Hatzilygeroudis 2018). The very deep convolutional neural network (VGG) architecture achieves 73.28% accuracy on the 8-class FER2013 dataset (Khaireddin and Chen 2021). Given the range of human emotion, however, many researchers prefer the AffectNet dataset (Mollahosseini, Hasani, and Mahoor 2017), which in addition to a classification target, further differentiates emotions along two continuous axes: valence and arousal. We discuss these values further in Section 3. The original model applied to AffectNet is AlexNet, obtaining an RMSE of 0.37 and 0.41 for valence and arousal, respectively (Mollahosseini, Hasani, and Mahoor 2017).

Since AlexNet's application to AffectNet, many improvements in the state-of-the-art have been made. BReG-NeXt (Hasani, Negi, and Mahoor 2020) replaces the short-

[*] These authors contributed equally.

cut bypass in ResNet (He et al. 2016) with a function with a bounded derivative to improve the gradient in back propagation. BReG-NeXt achieves state-of-the-art performance with valence RMSE of 0.2668 and arousal RMSE of 0.2482. Another architecture, the Visual Transformers with Feature Fusion (VTFF) model, can classify affect at 61.85% accuracy (Ma, Sun, and Li 2021), beating AlexNet's 58% accuracy. Furthermore, Li et al. (2021b) propose a mask vision transformer architecture and get an 8-class AffectNet classification accuracy of 64.57%. It appears, then, that transformer techniques can be effective for affect identification tasks. Thus we explore our own application of vision transformers further in Section 4.3.

We are not aware of any prior application of involution to affect identification, but the development of involution generally in prior research is described in Section 4.2 as we present our own involution-based model.

3 System Overview

The system consists of three major components:

Image Acquisition: Users press the "Capture" button to allow the system to take their picture. The system uses the Haar Cascade (Viola and Jones 2001) algorithm to detect and extract the face from the image.

Affect Identification: The image is then fed through one of several affect identification models. We consider existing models AlexNet (Mollahosseini, Hasani, and Mahoor 2017), ResNet (He et al. 2016), and Vision Transformer (ViT) (Dosovitskiy et al. 2020). We also consider two new models: one based on involution as applied in RedNet (Li et al. 2021a), and one using dynamic routing in vision transformers (Dosovitskiy et al. 2020; Sabour, Frosst, and Hinton 2017). These models are discussed in more detail further below. Every model outputs estimated valence and arousal values under the Russell circumplex model of affect (Russell 1980). Valence represents the level of negativity or positivity of an affect, while arousal represents the level of energy. Both values range from -1 to 1. Thus, the circumplex model is a valence-arousal coordinate system situating a range of emotions in 2-D space.

Music Recommendation: The system uses a 600k-song Spotify dataset (Chu and Roy 2017) as the music database for the recommendation system. Interestingly, this music database includes Spotify's determination of valence and arousal values for each song. There are larger Spotify datasets available, yet this one contains songs spread out widely on the valence-arousal dimensions. The system recommends the five songs in the dataset that are closest to the user's estimated valence and arousal, via nearest neighbor with Euclidean distance. The system includes the opportunity for user feedback, discussed further in Section 8.

4 Affect Identification

In this section we first describe the dataset used for training. We then discuss involution and dynamic routing in vision transformers, describing the architectures examined and situating them in the context of related work.

4.1 Data Description and Preprocessing

In affect identification, three major datasets exist: AffectNet (Mollahosseini, Hasani, and Mahoor 2017), Affect-in-the-wild (Zafeiriou et al. 2017), and FER2013 (Goodfellow et al. 2013). Among these, only AffectNet provides targets for both classification (8 affect classes) and regression (valence and arousal). Since we ultimately make use of both kinds of targets, we focus our work on AffectNet. Specifically, we use the publicly-released subset of AffectNet containing 287,651 training and 4,000 validation images. Since the AffectNet testing set has not been publicly released, we use their validation set as our testing set. Each RGB image is 224×224 pixels. Affect class frequency is unbalanced in AffectNet, ranging from 134,415 images of "happy", down to 3,750 images of "contempt".

4.2 Architecture 1: InvNet50

The affect identification models we've developed in this work are based on one of two approaches. In this section, we discuss the first approach, *involution*, which is a modification of convolution. Standard convolution layers allow a model to be *spatial-agnostic* (distinguishing features irrespective of location) and *channel-specific* (collecting features across various channels) (Li et al. 2021a). While these properties can allow translational equivariance (LeCun et al. 1998), the grouping of neighboring pixels reduces the ability to learn longer-distance spatial relationships. In addition, the number of channels grows in later layers of a deep CNN model. While this allows the model to capture some important features shared across channels, it also produces significant redundancy that increases computational cost (Jaderberg, Vedaldi, and Zisserman 2014).

Li et al. (2021a) propose involution to address these challenges through *spatial-specific* (aware of spatial relationships) and *channel-agnostic* (ignorant of channel-specific features) properties. In contrast to standard convolution, the involution kernel is not applied across local pixels. Rather, the kernel to be applied in a given location is determined dynamically based on the pixels at that location and other learned parameters. The learned parameters come from the image as a whole, in a mechanism similar to attention (Vaswani et al. 2017). This allows the model to more easily capture long-range spatial relationships of pixels. Furthermore, by sharing involution kernels along the channel dimension (channel-agnosticism), training of involution systems is more efficient without a significant loss in performance compared to a channel-specific approach.

We name our involution-based model InvNet50, due to some connections to ResNet50 (He et al. 2016) and ConvNeXt (Liu et al. 2022). Figure 1 and Table 1 outline the architecture, containing three main components: a stem layer, an inverted involution residual block, and a downsampling block. First, the stem layer compresses the image to remove irrelevant detail and reduce the computational requirements of the model. Similar to ResNet, we deploy a stem layer consisting of a 7×7 convolution, followed by a 2×2 max-pooling and a batch normalization layer. A stride of 2 is adopted in both the convolution and max-pooling layers

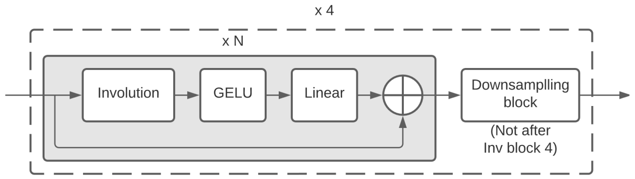


Figure 1: An illustration of the involution and downsampling blocks. In InvNet50, a sequence of four of these structures are preceeded by the stem layer and succeeded by the average pooling layer as described in Table 1.

Layer Name	Configuration
Stem layer	(Conv) 7×7, 64, stride 2 (Max Pooling) 2×2, stride 2
Inv block no.1 ($N = 3$)	(Inv) 7×7, 64, stride 1
Downsample block no.1	(Conv) 3×3, 128, stride 2
Inv block no.2 ($N = 4$)	(Inv) 7×7, 128, stride 1
Downsample block no.2	(Conv) 3×3, 256, stride 2
Inv block no.3 ($N = 6$)	(Inv) 7×7, 256, stride 1
Downsample block no.3	(Conv) 3×3, 512, stride 2
Inv block no.4 ($N = 2$)	(Inv) 7×7, 512, stride 1
	Average pool, 2d fc or 1d fc

Table 1: A brief overview of InvNet50 architecture. The number in the bracket after each Inv block represents the number of times the input goes through the Inv block after the stem or downsampling block. We omit the GELU and linear layers for brevity.

for consistency across the stem layer. Thus, the stem layer quickly downsamples the input's features by a factor of 4.

After the stem layer, we deploy a series of inverted bottleneck residual involution blocks based on ConvNeXt (Liu et al. 2022). Each block contains an involution layer, followed by layer normalization (Ba, Kiros, and Hinton 2016). For model configuration, we use an involution kernel of 7×7 and set the channels shared in a group to 16, thus reducing the number of parameters and computational cost without significantly harming the accuracy of the model (Li et al. 2021a). After that, we use Gaussian Error Linear Units (GELU) (Hendrycks and Gimpel 2016), dropout ($p = 0.5$), and a linear layer. At the end of each block, we deploy a batch normalization layer to reduce the internal covariate shift (Ioffe and Szegedy 2015) and therefore improve the convergence rate (Ioffe 2017). Lastly, to improve the generalization of the network we employ an element-wise addition ($\bigoplus$) between an identity layer and the output of the block (Simonyan and Zisserman 2015; He et al. 2016).

Between each of the four inverted involution residual blocks, we employ a downsampling block. Its purpose is similar to the stem layer, removing irrelevant details and encapsulating important features by increasing the number of output channels. In this case, we use a standard 2D convolution layer as our downsampling block. As done by Liu et al. (2022), we use a kernel size of 3 and a stride of 2 to quickly downsample the learned features. As done by He et al. (2016), we also adopt an adaptive average pooling layer and fully-connected layers before producing the output.

4.3 Architecture 2: DRViT

The second type of model we've developed for this work uses ideas from dynamic routing and vision transformers, thus we name it DRViT. One concept at the heart of the model is *self-attention* (Bahdanau, Cho, and Bengio 2015). For a given input, *query*, *key*, and *value* vectors are obtained via trained linear transformations. An add-multiply operation combines the query with all inputs' keys, the result is normalized, and softmax is applied to obtain the attention weights. Another multiply-add operation combines these with all value vectors to obtain the attention: a representation of the importance of every value in the sequence for the given query. More generally, attention is a representation of the importance of every item in the input to a particular item. The above calculations are summarized:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \operatorname{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}$$

for matrices **Q**, **K**, and **V** of all query, key, and value vectors, respectively, and d_k the dimensionality of a key vector.

Transformers were first applied to natural language processing tasks (Vaswani et al. 2017). The input text is treated as a sequence of words, word embeddings are computed to convert the text to numerical input, positional encoding is applied to capture the sequential nature of the text, and finally self-attention is applied. More specifically, transformers use *multi-head* attention, in which the query, key, and value vectors are divided into n components called *heads* via another linear transformation. n self-attention operations are run in parallel, one on each head. Each head ends with a feed-forward neural network, adding a non-linear transformation and further parameterizing the head for training. Finally, the output of each head is concatenated to obtain the output of multi-head attention.

Dosovitskiy et al. (2020) introduced the first largely successful application of transformers to image tasks with the Vision Transformer (ViT). In this approach, images are split into patches and flattened before proceeding with positional encoding and the remaining transformer steps.

Sabour, Frosst, and Hinton (2017) note that the feed forward neural network layers in each head cannot learn the hierarchical structure of image features and are not rotationally equivariant. Thus, they propose *dynamic routing*, which we make use of in the dynamic routing vision transformer (DRViT). Here, the feed forward neural network layer in each head is replaced with a dynamic routing algorithm.

In dynamic routing, the idea of a classic neuron is extended to a *capsule*, which outputs an *activity vector* of values instead of a single value. The magnitude of the vector represents the estimated probability of some detected feature, while the components represent properties of that feature—potentially including spatial properties that can be missed in convolutional models.

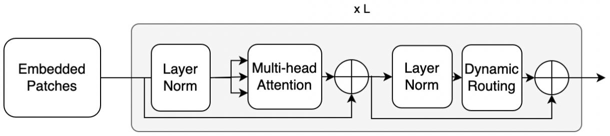


Figure 2: An illustration of the Dynamic Routing for Vision Transformers (DRViT) architecture.

To sketch the dynamic routing process, let $\mathbf{u}_i$ be activity vector i from the previous layer of capsules. Capsule j in the next layer outputs $\mathbf{v}_j$ where:

$$\hat{\mathbf{u}}_{j|i} = \sum_i \mathbf{W}_{ij} \times \mathbf{u}_i \qquad \mathbf{s}_j = \sum_i c_{ij}\hat{\mathbf{u}}_{j|i}$$

$$\mathbf{v}_j = \frac{||\mathbf{s}_j||^2}{1 + ||\mathbf{s}_j||^2} \frac{\mathbf{s}_j}{||\mathbf{s}_j||}$$

Note that $\mathbf{v}_j$ is a squashed value such that $\mathbf{v}_j \rightarrow \mathbf{0}$ for small $||\mathbf{s}_j||$ and $\mathbf{v}_j \rightarrow \mathbf{1}$ for large $||\mathbf{s}_j||$. $\mathbf{W}_{ij}$ represents trainable weight matrices. Finally, c_{ij} is the *coupling coefficient*, representing the importance of capsule i on capsule j. These c_{ij} are obtained via an iterative process, in which c_{ij} values based on $\hat{\mathbf{u}}_{j|i} \cdot \mathbf{v}_j$ are computed and used to calculate new $\mathbf{s}_j$ and $\mathbf{v}_j$, which then lead to new c_{ij}, typically through 3 iterations until a final $\mathbf{v}_j$ value is output. See the work of Sabour, Frosst, and Hinton (2017) for complete details. In short, this approach allows for the recognition of spatial relationships more effectively than convolutional models, due to the activity vectors capturing not just feature presence but also features' spatial relationships.

Song et al. (2021) propose a *dynamic grained encoder* for vision transformers and obtain good results in ImageNet classification. With this inspiration, we also apply dynamic routing to vision transformers, but place the routing layer after the multi-head attention function instead of after the patches of images as done by Song et al. A system diagram is provided in Figure 2. We hypothesize that this alternative can further help the system learn features that focus on the most important parts of the image. In our experiments with DRViT, we consider only three encoder blocks (L). The number of heads is 8, and the dimension of embedding layers is 256. In our experiments with DRViT, we use an embedding dimension of 256, $L = 3$ encoder blocks, and 8 heads per block. Our last layer is a feed-forward perceptron.

5 Experimental Setup

In this section we describe the variables explored in our affect identification model experiments: the model architecture, whether or not data augmentation is applied, and what output the model is tasked to provide.

We prepare these experiments with the intent of comparing them first to the AlexNet system of Mollahosseini, Hasani, and Mahoor (2017). This architecture was originally applied to general image recognition (Krizhevsky, Sutskever, and Hinton 2012) in the ImageNet dataset (Deng 2009). Mollahosseini, Hasani, and Mahoor use the same architecture, but without transfer learning, trained on the AffectNet dataset. The model contains five convolution layers, each followed by max-pooling and batch normalization layers, and finally three fully-connected layers. Mollahosseini, Hasani, and Mahoor actually create two separate copies of AlexNet, with one trained for valence estimation, and one for arousal. We use their reported results on AffectNet as a baseline comparison for our own work.

For further comparison to existing work, we also consider two well-known models: ViT and ResNet50. Vision Transformer (ViT) (Dosovitskiy et al. 2020) applies multi-head attention to focus on important components of the image. ResNet50 (He et al. 2016) is a convolutional neural network containing 50 layers that uses residual techniques to avoid overfitting and vanishing gradients. We choose ResNet50 over other variants of the Residual Network since it is the most popular one, given its balance between number of parameters and accuracy. Both ViT and ResNet50 were originally trained on ImageNet, and we used transfer learning to further train the parameters on the AffectNet dataset.

For InvNet50 and DRViT, our experiments consider three different training sets: a non-augmented set, a set augmented under plan A, and another augmented under plan B. We explore these plans because some preliminary experiments showed different models had different augmentation preferences. The non-augmented set is the original AffectNet dataset with unbalanced class counts. Augmentation plans A and B have much in common. For affect classes with more than 20,000 images, we randomly select 20,000 images. For classes with fewer, we select all available images. For each selected image, one of the following is applied with equal probability: no augmentation; Gaussian blur (Gedraite and Hadad 2011) to blur the image using Gaussian-generated noise; horizon flip (Lei et al. 2019) to horizontally flip the image; color jitter (Hou, Zheng, and Gould 2020) to randomly adjust brightness, hue, saturation, and contrast; or random erasing (Zhong et al. 2017) to erase a rectangular region of the image. This augmentation is done "online", in which the augmentation occurs during training with potentially different random choices for each original selected image in each epoch. While this approach does lead to variability in the dataset across experiments (even within the same augmentation plan), it has been found to bring useful diversity to the training set (Cubuk et al. 2019).

Augmentation plans A and B consider in different ways the balance in class counts and the similarity to the non-augmented dataset. Plan A has perfect balance of class counts while more significantly adjusting the nature of the dataset. Plan B makes smaller adjustments to reduce the class count skew but does not achieve balance. More precisely, in plan A we consider each affect class 20,000 times. For a given affect class, an image is randomly selected: 1 of 20,000 for larger classes, or 1 of fewer images for smaller classes. Augmentation is applied to each selected image as described above. Thus, the resulting number of images for plan A is 160,000 (20,000 × 8 classes), and the class counts are balanced. In contrast, plan B considers each image once, but, as described above, caps larger classes at 20,000 images. Augmentation is applied to each selected image as described above. Thus, by plan B, some affect classes still have fewer observations than others, but the skew is reduced. Under plan B, the total number of observations is 108,021 images. Again, both plans use online augmentation, and so any random selections are made anew each epoch.

We name the final variable of our experiments *models* × *outputs*, with possible values of 2×1 and 1×2. With 2×1, we refer to two copies (submodels) of the same architecture, each with one output: one submodel trained to predict valence, and one arousal. Recall that this is the approach used by Mollahosseini, Hasani, and Mahoor (2017) in AlexNet. The alternative, 1×2, refers to one model trained to produce two outputs: one for valence and one for arousal. This is the approach of Toisoul et al. (2021).

In training, both DRViT and InvNet50 use L2 loss and a batch size of 64. We use Adam optimization with a decoupled weight decay of 0.05 and a learning rate of 0.001 (Loshchilov and Hutter 2017). To avoid overfitting, we apply various approaches including early stopping and learning rate reduction schedulers based on the validation error and the number of epochs (Gençay and Qi 2001; Mollahosseini, Hasani, and Mahoor 2017; Toisoul et al. 2021). Experiments with SGD instead of Adam optimization led to slightly worse results, thus we do not consider it further.

As in prior affect identification work (Mollahosseini, Hasani, and Mahoor 2017; Hasani, Negi, and Mahoor 2019; Weiler, Hamprecht, and Storath 2018), we use 4 different performance metrics for valence and arousal. Consider first the Root Mean Square Error (RMSE) measure, which computes the average distance between each predicted value $\hat{y}_i$ and ground truth value y_i. RMSE can be strongly affected by outliers (Bermejo and Cabestany 2001), and so Mollahosseini, Hasani, and Mahoor (2017) propose the additional use of the Pearson Correlation Coefficient (CORR) and Concordance Correlation Coefficient (CCC). Briefly, CORR measures the linearity of the $\mathbf{y}$ versus $\hat{\mathbf{y}}$ relationship, while CCC measures the numerical agreement between the values.

Finally, Mollahosseini, Hasani, and Mahoor (2017) argue that in valence and arousal an agreement in sign (positive or negative) between y_i and $\hat{y}_i$ can be more important than some differences in magnitude. Thus, we also use the sign agreement (SAGR) metric, which outputs 1 or 0 for sign agreement or disagreement, respectively.

6 Experimental Results and Analysis

Transfer Learning: Considering the results in Table 2, we first compare the AlexNet baseline (row 1) to the transfer learning models: ResNet50 (rows 2 and 3) and ViT (rows 4 and 5). In prediction of valence, AlexNet has better results in all four metrics than ResNet50 and ViT. For arousal, AlexNet has slightly worse performance than ResNet50 and ViT. These patterns hold whether we use no augmentation or augmentation A. Additional experiments might bring more insights on ResNet50 and ViT, but given their failure to significantly improve upon AlexNet in these first experiments, we instead move on to other experiments.

Model × Outputs: We next compare the 1×2 and 2×1 designs for InvNet50. Our intuitions on this matter go in both directions. On the one hand, the 2×1 approach allows each submodel to focus on only a single output, thus the parameters can be attuned to just that output. On the other hand, while valence and arousal are distinct values, it also seems intuitive that there are correlations between them. With the 1×2 approach of training a single model to output both values, the model might leverage these relationships. Given this apparent close trade-off, we run experiments on both approaches.

For InvNet50 with no augmentation (rows 6 and 8), we find similar results between 1×2 and 2×1, perhaps with a slight preference for 1×2. For augmentation B (rows 7 and 10), this pattern continues. We conclude that 1×2 is only slightly preferable. Apparently, our hypothesis about the close trade-off between the two approaches was accurate. 1×2 becomes much more attractive, however, when we also consider that 2×1 (two models) requires nearly twice as much training time and has nearly double the parameters. We therefore focus on 1×2 in subsequent experiments.

Augmentation: Next, we consider augmentation plans, focusing first on 1×2 for InvNet50 (rows 8, 9, and 10). Plan A gives better results than no augmentation in this experiment. Plan B results, however, are superior to A and significantly superior to no augmentation. In fact, this pattern is found in the 2×1 design as well (rows 6 and 7). It seems safe to conclude, then, that plan B is the best augmentation choice for InvNet50. This is an interesting result, since augmentation B results in a mere 108k images, compared to 160k for plan A and 288k for the original AffectNet. It seems likely that InvNet50 is not harmed by this smaller dataset due to its significantly smaller number of parameters compared to AlexNet (10.5M versus 58.2M per model); InvNet50 with augmentation B (rows 7 and 10) beats AlexNet in nearly every measure. An interesting question for future work is to examine the effect of augmentation plans like A and B, but modified to have more images.

Similarly, we consider augmentation plans for 1×2 DRViT models (rows 11, 12, and 13). We see that plan A achieves just slightly worse performance overall compared to no augmentation, while plan B is clearly worse. Given InvNet50's preference for plan B, it is surprising that B is the worst of all for DRViT, and that no augmentation is preferred. Again, this might be explained by the difference in dataset sizes among the three plans. Perhaps some aspects of DRViT's architecture is more data-hungry than InvNet50's approach; we consider this further below. Regardless, the fact that DRViT performs best with no augmentation in our tests means the model can avoid that extra training cost.

Architectures: Thus, our best result is DRViT 1×2 with no augmentation (row 11). Our best InvNet50 result is 1×2 with augmentation B (row 10). Both of these systems show stronger results than the AlexNet baseline, particularly in arousal. This is an interesting outcome given the significantly lower parameter counts for both InvNet50 and DRViT compared to AlexNet, ResNet50, and ViT. These results match our intuition. First, both InvNet50 and DRViT use an attention mechanism. In affect identification, facial images contain some regions of particular importance (e.g., mouth, eyes), and so it is reasonable to expect that attention helps a model focus on the most important regions.

					Valence				Arousal			
			Params		RMSE	CORR	CCC	SAGR	RMSE	CORR	CCC	SAGR
ID	Arch	M × O	(M)	Aug	(<)	(>)	(>)	(>)	(<)	(>)	(>)	(>)
1	AlexNet	2 × 1	2 × 58.2	No	0.37	0.66	0.60	0.74	0.41	0.54	0.34	0.65
2	ResNet50	2 × 1	2 × 25.0	No	0.41	0.58	**0.53**	**0.68**	0.43	0.46	**0.47**	0.65
3	ResNet50	2 × 1	2 × 25.0	*A*	**0.39**	**0.59**	**0.53**	0.67	**0.40**	**0.48**	0.41	**0.66**
4	ViT	2 × 1	2 × 85.0	No	0.40	**0.58**	0.55	**0.66**	0.42	0.50	**0.46**	0.62
5	ViT	2 × 1	2 × 85.0	*A*	**0.39**	0.57	**0.56**	0.65	**0.39**	**0.52**	0.41	**0.68**
6	InvNet50	2 × 1	2 × 10.5	No	0.43	0.57	0.53	0.72	0.36	0.50	0.43	0.75
7	InvNet50	2 × 1	2 × 10.5	*B*	0.37	0.63	0.61	0.76	0.34	0.53	0.49	0.78
8	InvNet50	1 × 2	10.5	No	0.42	0.59	0.55	0.73	0.36	0.51	0.45	0.74
9	InvNet50	1 × 2	10.5	*A*	**0.36**	0.62	0.57	**0.77**	**0.33**	0.51	0.42	0.79
10	InvNet50	1 × 2	10.5	*B*	0.37	**0.65**	**0.63**	**0.77**	**0.33**	**0.55**	**0.52**	**0.80**
11	DRViT	1 × 2	13.0	No	**0.36**	**0.68**	**0.66**	0.78	0.36	**0.67**	**0.53**	0.75
12	DRViT	1 × 2	13.0	*A*	0.37	0.66	0.63	**0.79**	**0.35**	0.65	0.48	**0.77**
13	DRViT	1 × 2	13.0	*B*	0.39	0.61	0.57	0.72	0.37	0.56	0.48	0.63

Table 2: Results for various model experiments. Each row includes the architecture (Arch), the *model*×*outputs* value (M×O), the number of parameters in millions (Params (M); 2× when M×O = 2 × 1), and the type of augmentation used (Aug). Each row is completed with values for the four metrics in both valence and arousal prediction. Each measure is labeled (<) or (>) to indicate lower or higher numbers are better, respectively. For each architecture, the best value for each measure is in bold.

We hypothesize that InvNet50 suffers slightly compared to DRViT because its involution kernel includes only some characteristics of attention rather than the entire attention mechanism as traditionally defined. In addition, for simplicity in InvNet50, we only consider GELU activation and linear layers, similar to ViT. In contrast, DRViT uses dynamic routing to capture hierarchical relationships of features extracted from multi-head attention. This may be the source of the previously hypothesized higher data requirements of DRViT compared to InvNet50, but the explanation for the different behavior remains uncertain.

While our models do not outperform the state-of-the-art on AffectNet, our improvement over AlexNet shows promise for further development. The precise reasons for DRViT's stronger performance, and how both approaches may be further improved, is a matter of future study.

7 Music Recommender

Each of the previously described affect identification models output v_i and a_i, corresponding to valence and arousal for the input image, respectively. Recall that each of these are in the range $[-1, 1]$. These outputs serve as input into the music recommender. The recommender uses a 600k-song Spotify dataset (Chu and Roy 2017), with each song already labeled with values in the range $[0, 1]$ for valence (v_{sp}) and "energy" (e_{sp}). Energy is analogous to arousal, and so we obtain valence v_s and arousal a_s values in $[-1, 1]$ for each song via $v_s = 2 \cdot v_{sp} - 1$ and $a_s = 2 \cdot e_{sp} - 1$. Given these values, we can compute the Euclidean distance $d((v_i, a_i), (v_s, a_s))$ on the valence-arousal plane between an image i and a song s. With this distance measure, we use 5-nearest neighbor to obtain the five songs most similar to the identified affect.

Our complete system (affect identifier and music recommender) is deployed as a web application (see https://github.com/anhphuongdo34/eaai23-client-dup and https://github.com/anhphuongdo34/eaai23-server-dup). The front-end of the app was built using the React framework, while the back-end was written using Python and the Flask framework and hosted on Google Cloud Platform. The app acquires access to the user's front-facing camera or webcam to capture a photo of their face. It then uses OpenCV to extract the cropped facial image to feed through the model for the prediction of valence and arousal.

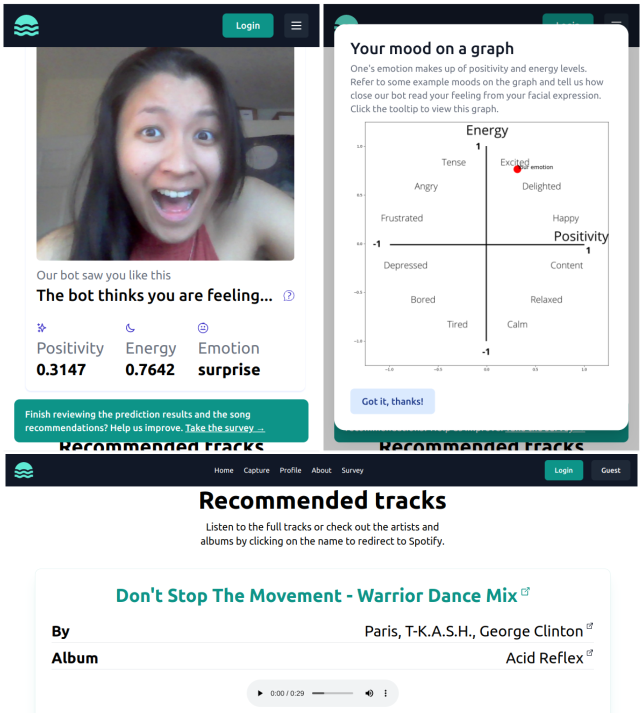
The app displays the valence-arousal value of the image as a point on a circumplex graph, and indicates 1 of 8 classifications corresponding to the (v_i, a_i) point, such as "happy" or "sad". To provide this classification, we use a decision tree trained on the AffectNet dataset, using valence and arousal as inputs and the classification as the target. We tuned the hyperparameters of the tree using a grid search, ultimately using a maximum depth of 15, a maximum of 15 samples per node, and at least 5 samples needed to split a node. This simple approach achieves 96% accuracy on the AffectNet validation set (used as the testing set, just as for the affect identifier models).

The app then displays the top five closest-matching songs, determined as described above. The user may preview the first 30 seconds of a song, or listen to it in its entirety on Spotify through a provided link. Figure 3 shows an example of the experience.

8 User Experience Surveys

To gain insight into the user experience and preferences, we conducted user testing of the system. For these tests, the app uses 1×2 plan *B* InvNet50, rather than the slightly superior DRViT, since the DRViT models have memory requirements beyond the modest resources of the app host. We invited users of the system to fill out an IRB-approved survey, rating the following statements on a 5-point Likert scale from 1 (strongly disagree) to 5 (strongly agree).

1. I prefer listening to music that matches my mood closely.



Our bot saw you like this

The bot thinks you are feeling...

Positivity 0.3147

Energy 0.7642

Emotion surprise

Finish reviewing the prediction results and the song recommendations? Help us improve. Take the survey →

Your mood on a graph

One's emotion makes up of positivity and energy levels. Refer to some example moods on the graph and tell us how close our bot read your feeling from your facial expression. Click the tooltip to view this graph.

Recommended tracks

Listen to the full tracks or check out the artists and albums by clicking on the name to redirect to Spotify.

Don't Stop The Movement - Warrior Dance Mix

By Paris, T-K.A.S.H., George Clinton

Album Acid Reflex

Figure 3: The system demo with the predicted valence-arousal, emotion and the song recommendations.

2. I prefer listening to music that helps improve my mood (i.e., calm me down when I am angry, cheer me when I am sad, etc.)
3. Most of the time, my emotion falls into one and only one of the listed categories without overlapping.
4. If possible, I wanted to adjust the emotion before getting the song recommendations.

Responses to the first two statements give insight about a core assumption: that users want to listen to music that matches their mood (question (1)), as opposed to music aiming to change their mood (question (2)). Responses to the third statement can suggest a user preference for classification versus a (v_i, a_i) point on a circumplex plane. We anticipate that feelings are usually complex, and are better represented within a continuous range instead of by a discrete class. Responses to the final statement give insight into users' perceptions of affect identification accuracy.

Summary results of 39 responses to the user survey are shown in Table 3. Results show that users slightly prefer listening to music that matches their mood, rather than music aimed at changing their mood (mean 4.03 for question 1, versus 3.66 for question 2). Responses to question 3 are near the middle with a 3.13 average. That is, users have no strong opinion about whether their emotions fall into exactly one or multiple classes. Perhaps, then, the flexibility of both a classification and (valence, arousal) measure is more useful. Responses to question 4 again are near the middle with a 3.14 average. This suggests that users are ambivalent about whether the system is accurately identifying their emotion.

Item	Average	Median	Std. Dev.
1	4.03	4	1.05
2	3.66	4	1.15
3	3.13	3	1.19
4	3.14	3	1.29

Table 3: Results of the user experience survey.

Given the complexity of emotions, the range of possibilities in the circumplex model, and the error in the tested models (depsite success over the AlexNet baseline), such a result is perhaps not surprising. It is noteworthy, at least, that the rating is not even lower.

9 Conclusions and Future Work

In this paper, we first compare several deep learning models for affect identification. We find that DRViT is most effective, combining dynamic routing with vision transformers. We also find that the involution-based InvNet50 model is quite effective. Both models yield better results than AlexNet in all metrics, with far fewer parameters. The output of an affect identifier then serves as the input into a music recommendation system. We deploy a web application in which music is recommended based on the perceived affect of the user. Thus, by understanding the user's affect, our system aims to provide emotion-aware music recommendation.

The work described in this paper can be extended in many ways. To consider a few, first note that more combinations of experimental variables may bring further insight on the architectures, model × output designs, and augmentation plans. More fine-grained experiments could be considered as well: for example, more architecture variables or augmentation plans that lead to datasets of varying size—particularly sizes closer to the original AffectNet dataset.

The system as designed aims to recommend songs that match the predicted affect. One might consider an alternative, in which the system aims to gradually move the user into a positive mood through a sequence of songs traversing the circumplex plane in a positive direction.

Finally, the song valence and arousal values from the Spotify database were taken as truth. In reality, though, these values are subject to error. Consideration of valence and arousal estimation for *songs*, as opposed to merely faces as in this work, could bring further improvements to the user experience. Other song attributes included with the Spotify database (e.g., danceability, liveness, tempo) may also provide insight into this kind of song classification.

Acknowledgments

We are grateful for the support of the Tenzer Center, the J. William Asher and Melanie J. Norton Endowed Fund in the Sciences, and the Kranbuehl, Roberts, and Hillger Endowed Fund for Faculty Summer Research.

References

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer Normalization. *Stat.ML*, 1050: 21.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. *Proceedings of the International Conference on Learning Representations (ICLR'15).*

Bermejo, S.; and Cabestany, J. 2001. Oriented principal component analysis for large margin classifiers. *Neural Networks*, 14(10): 1447–1461.

Chu, E.; and Roy, D. 2017. Audio-visual sentiment analysis for learning emotional arcs in movies. In *2017 IEEE International Conference on Data Mining (ICDM)*, 829–834. IEEE.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2019. RandAugment: Practical data augmentation with no separate search. *CoRR*, abs/1909.13719.

Deng, J. 2009. A large-scale hierarchical image database. *Proc. of IEEE Computer Vision and Pattern Recognition, 2009*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. *CoRR*, abs/2010.11929.

Gedraite, E. S.; and Hadad, M. 2011. Investigation on the effect of a Gaussian Blur in image filtering and segmentation. In *Proceedings ELMAR-2011*, 393–396.

Gençay, R.; and Qi, M. 2001. Pricing and hedging derivative securities with neural networks: Bayesian regularization, early stopping, and bagging. *IEEE Transactions on Neural Networks*, 12(4): 726–734.

Giannopoulos, P.; Perikos, I.; and Hatzilygeroudis, I. 2018. Deep learning approaches for facial emotion recognition: A case study on FER-2013. In *Advances in hybridization of intelligent methods*, 1–16. Springer.

Goodfellow, I. J.; Erhan, D.; Carrier, P. L.; Courville, A.; Mirza, M.; Hamner, B.; Cukierski, W.; Tang, Y.; Thaler, D.; Lee, D.-H.; et al. 2013. Challenges in representation learning: A report on three machine learning contests. In *International conference on neural information processing*, 117–124. Springer.

Hasani, B.; Negi, P. S.; and Mahoor, M. 2020. BReG-NeXt: Facial affect computing using adaptive residual networks with bounded gradient. *IEEE Transactions on Affective Computing*.

Hasani, B.; Negi, P. S.; and Mahoor, M. H. 2019. Bounded residual gradient networks (breg-net) for facial affect computing. In *2019 14th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, 1–7. IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; and Gimpel, K. 2016. Bridging Nonlinearities and Stochastic Regularizers with Gaussian Error Linear Units. *CoRR*, abs/1606.08415.

Hou, Y.; Zheng, L.; and Gould, S. 2020. Learning to structure an image with few colors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10116–10125.

Hsu, K.; Chou, S.; Yang, Y.; and Chi, T. 2016. Neural Network Based Next-Song Recommendation. *CoRR*, abs/1606.07722.

Hu, Y.; and Ogihara, M. 2011. NextOne Player: A Music Recommendation System Based on User Behavior. In *ISMIR*, volume 11, 103–108.

Ioffe, S. 2017. Batch renormalization: Towards reducing minibatch dependence in batch-normalized models. *Advances in neural information processing systems*, 30.

Ioffe, S.; and Szegedy, C. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR*, abs/1502.03167.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Speeding up Convolutional Neural Networks with Low Rank Expansions. *CoRR*, abs/1405.3866.

James, H. I.; Arnold, J. J. A.; Ruban, J. M. M.; Tamilarasan, M.; and Saranya, R. 2019. Emotion based music recommendation system. *IRJET*, 6(03).

Ji, K.; Sun, R.; Shu, W.; and Li, X. 2015. Next-song recommendation with temporal dynamics. *Knowledge-Based Systems*, 88: 134–143.

Juslin, P. N.; Sloboda, J. A.; et al. 2001. Music and emotion. *Theory and research.*

Khaireddin, Y.; and Chen, Z. 2021. Facial Emotion Recognition: State of the Art Performance on FER2013. *CoRR*, abs/2105.03588.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11): 2278–2324.

Lei, C.; Hu, B.; Wang, D.; Zhang, S.; and Chen, Z. 2019. A preliminary study on data augmentation of deep learning for image classification. In *Proceedings of the 11th Asia-Pacific Symposium on Internetware*, 1–6.

Li, D.; Hu, J.; Wang, C.; Li, X.; She, Q.; Zhu, L.; Zhang, T.; and Chen, Q. 2021a. Involution: Inverting the Inherence of Convolution for Visual Recognition. *CoRR*, abs/2103.06255.

Li, H.; Sui, M.; Zhao, F.; Zha, Z.; and Wu, F. 2021b. MViT: Mask Vision Transformer for Facial Expression Recognition in the wild. *CoRR*, abs/2106.04520.

Liu, Z.; Mao, H.; Wu, C.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022. A ConvNet for the 2020s. *CoRR*, abs/2201.03545.

Logan, B. 2004. Music Recommendation from Song Sets. In *ISMIR*, 425–428.

Loshchilov, I.; and Hutter, F. 2017. Fixing Weight Decay Regularization in Adam. *CoRR*, abs/1711.05101.

Ma, F.; Sun, B.; and Li, S. 2021. Facial Expression Recognition with Visual Transformers and Attentional Selective Fusion. *IEEE Transactions on Affective Computing*.

Mollahosseini, A.; Hasani, B.; and Mahoor, M. H. 2017. AffectNet: A Database for Facial Expression, Valence, and Arousal Computing in the Wild. *IEEE Transactions on Affective Computing*, PP(99): 1–1.

Russell, J. A. 1980. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6).

Sabour, S.; Frosst, N.; and Hinton, G. E. 2017. Dynamic routing between capsules. *Advances in neural information processing systems*, 30.

Samuvel, D. J.; Perumal, B.; and Elangovan, M. 2020. Music recommendation system based on facial emotion recognition. *Publicado en 3C Tecnología. Special Issue*.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *International Conference on Learning Representations*.

Song, L.; Zhang, S.; Liu, S.; Li, Z.; He, X.; Sun, H.; Sun, J.; and Zheng, N. 2021. Dynamic grained encoder for vision transformers. *Advances in Neural Information Processing Systems*, 34: 5770–5783.

Taruffi, L.; Pehrs, C.; Skouras, S.; and Koelsch, S. 2017. Effects of sad and happy music on mind-wandering and the default mode network. *Scientific reports*, 7(1): 1–10.

Toisoul, A.; Kossaifi, J.; Bulat, A.; Tzimiropoulos, G.; and Pantic, M. 2021. Estimation of continuous valence and arousal levels from faces in naturalistic conditions. *Nature Machine Intelligence*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. *CoRR*, abs/1706.03762.

Viola, P.; and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, I–I.

Weiler, M.; Hamprecht, F. A.; and Storath, M. 2018. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 849–858.

Zafeiriou, S.; Kollias, D.; Nicolaou, M. A.; Papaioannou, A.; Zhao, G.; and Kotsia, I. 2017. Aff-wild: valence and arousal'In-the-Wild'challenge. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 34–41.

Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; and Yang, Y. 2017. Random Erasing Data Augmentation. *CoRR*, abs/1708.04896.