Emosic: Image driven affective computing on mobile devices

Charlie Hewitt
Computer Lab, University of Cambridge
Cambridge, United Kingdom

***Abstract*— This paper presents a number of convolutional neural network architectures for affect determination on mobile devices, these networks are designed with the goal of retaining high performance while minimising storage requirements. A comparative evaluation of these architectures is carried out on a very large, in-the-wild dataset of facial images achieving an accuracy of 58% for eight-class emotion prediction and average RMSE of 0.39 for valence/arousal prediction, similar results to the dataset baseline. To demonstrate the feasibility of real world applications, the trained models are converted for use on the iOS platform and deployed within an application providing music recommendations based on predicted user affect. A user study is conducted to assess the performance of the models in this context, achieving similar prediction performance as for the dataset with an accuracy of 54% and average RMSE of 0.30, also achieving a positive user rating for song recommendations.**

I. INTRODUCTION

A. Motivation

Affective computing has historically remained confined to laboratory settings, typically only involving small studies and with little in the way of large-scale practical application. Recent advances in deep machine learning techniques and increasing availability of large, in-the-wild datasets have lead to improved performance in affect recognition tasks such as prediction of emotion, valence and arousal from facial images in real world scenarios, not just in constrained environments. The ubiquitousness of mobile devices with advanced sensors, including high quality cameras, means that the application of affective computing technologies to end-user applications is now a real possibility.

This paper aims to explore the feasibility of obtaining state-of-the-art emotion recognition from image performance using machine learning approaches within the constrained environment of a mobile device, as well as how readily the output of these models can be used within a mobile application.

The Emosic application prompts the user to take an image of their face and predicts the prominent emotion—neutral, happy, sad, surprised, afraid, disgusted, angry or contemptuous—as well as the valence and arousal of the subject using convolutional neural network (CNN) models, as described in Sec. II. A number of recommended songs are then presented to the user based on the predicted user affect.

The application is intended as a proof-of-concept that emotionally intelligent user interfaces (EIUI) on mobile devices are now feasible using modern machine learning approaches and large, in-the-wild datasets. A more detailed description of the application implementation, along with screenshots, is given in Sec. III. Conclusions and suggestions of future work are provided in Sec. IV, with a summary of this paper's contributions and related work making up the remainder of Sec. I. The source code for the application and machine learning setup is available on GitHub [13].

B. Contributions

This research has two primary contributions:

- Comparative evaluation of three CNN architectures aimed for deployment to mobile devices, constrained primarily by the storage size required for the model, trained for emotion and valence/arousal prediction in the wild using the newly available AffectNet dataset [25].
- Demonstration of deployability of these pre-trained CNNs for applied affective computing in a mobile application, in this case for the task of song recommendation.

C. Related Work

Commercially available tools for real world affect determination are fairly limited. Affectiva [22] is the most established offering with a number of successful applications, for example in adaptive children's computer games, automatic tagging on the Imgur image hosting site and assessing viewer reception of television adverts. Microsoft are also trialing Emotion API [24] which offers similar functionality, though has so far seen little in terms of real world applications.

Small-scale deployments of affect recognition models have generally focussed on video games [21], medical applications [33], [23] and driver emotion [1]. There has so far been very little development of EIUIs, perhaps this is in part due to user reluctance based on privacy concerns as well as technological limitations.

Mobile affective computing has mostly remained limited to activity monitoring based on accelerometer data and call, SMS and application usage with only two examples involving the camera as of 2017 [30].

The recently released AffectNet dataset [25] is a very large scale (450,000 images) in-the-wild annotated dataset for training of affective computing models. It includes annotations for 8 emotion categories, valence and arousal on a continuous scale from -1 to 1 and facial bounding boxes and landmarks. Previous in-the-wild dataset were generally smaller and did not include annotations of valence and arousal; FER-2013 [10] included 35,000 images with 7 emotion categories, FER-Wild [26] included 25,000 similarly annotated images and EmotioNet [6] contained 100,000 images with 23 emotion categories. The increased availability of these large annotated libraries of facial images enables far more productive research in the field of affective computing.

Existing machine learning approaches typically don't consider model size as an important attribute in architecture design. General image classification architectures such as InceptionV3 [35], ResNet50 [12] and VGG16 [32] result in data files of significant size; 90MB, 97MB and 528MB respectively. The CNNEmotions architecture [19] designed for the task of emotion classification is certainly too large for any realistic mobile application (475MB), as is VGGFace [29]. The only architecture specifically designed with mobile deployment in mind is MobileNet [14] which, at 16.4MB, is certainly reasonable for mobile deployment[1].

Implementations of lighter weight CNN architectures for affective computing have focussed primarily on real-time classification, though often produce smaller models as a side effect of this. [5], [31] and [9] specify models that achieve quite high classification accuracy ($\sim$60% on FER2013) for frames in video feeds in real time, with file sizes generally smaller than 30MB.

II. AFFECT RECOGNITION METHOD

Given the goal of mobile deployment the final model size must remain reasonable for inclusion in a mobile application; Google and Apple both impose limits on app size in the Play Store and AppStore respectively. For installation over cellular network Apple limits apps to 150MB (100MB before Sep 2017) and all apps Google imposes a limit of 100MB (50MB before Sep 2015). Given that any models included within an app likely only augment the primary use case, rather than provide the use case itself (e.g. EIUI), the storage space used by these models should remain well under this 100MB limit.

Cloud offload is perhaps an obvious solution to the issue of constrained resource, but for this application facial imagery of the user is unavoidably involved so privacy becomes an immediate concern. Because of this, as well as concerns regarding latency, local execution is the preferred course of action.

Consequently a goal model size of 15MB is imposed for this research; as two models are required (one for emotion classification and one for valence/arousal regression) the total contributed file-size should remain less than 30MB, meaning the app will be approximately 50MB in size overall. The time and computational resource taken to obtain predictions from images are also a factor to consider on mobile devices, though given the simplicity inherent in architectures of this size is of little issue for any of the models presented in this paper.

Three CNN architectures are explored based on previously established networks; a design similar to AlexNet [18] using a series of convolution layers with incrementally smaller kernels interspersed with max-pooling layers, an architecture based on VGG16 [32] with stacked 3×3 convolution layers interspersed with max-pooling layers and a network based on MobileNet [14] utilising depth-wise separable convolutions to maximise efficiency.

[1]All sizes relate to pre-trained CoreML [2] models available from CoreML Store [36]

All CNNs are implemented using Keras [7] and trained on an NVIDIA GeFore GTX 1080 Ti GPU using TensorFlow [11].

A. Preprocessing and Training Procedure

The AffectNet dataset [25] contains images of a highly heterogeneous nature, in order to produce suitable images for input to a CNN the faces are cropped and resized to 128×128 pixels. AffectNet's provided facial bounding box annotations are used for this purpose.

Only manually annotated images are used[2]; for emotion classification all images annotated with invalid emotions (8: none, 9: uncertain and 10: no-face) are discarded leaving a total training set of 287,651 images and a validation set of 4000 images. For valence/arousal regression all images with invalid annotations, indicated using a value of -2, are discarded leaving a training set of 320,739 images and a validation set of 4500 images.

Weighted-loss is used for emotion classification to account for the imbalance in the training set as this achieved the best results (compared with up- and down-sampling) in [25]. For valence/arousal regression data imbalance is again a problem resulting in over-fitting and potentially reduced performance; the mean annotations of the training set are 0.19 and 0.09 for valence and arousal respectively, while for the validation set are -0.16 and 0.30. Attempting to rectify this by down-sampling did little to improve performance so the full training set is used.

The Adam optimiser [16] is used throughout with suggested parameters $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$, this is due to its design focus for machine learning task on large datasets.

Randomised data augmentation is used for the training set with potential for images to be rotated by up to 20 degrees, translated by up to 10% (in both x- and y-directions) and flipped in the x-direction. All image data is normalised from $[0, 255]$ to $[0, 1]$ to increase the speed of training.

Batch size is maximised in order to best encapsulate the varied nature of the data and therefore improve training; 400 for architectures 1 and 2, and 250 for architecture 3, limited by available memory on the training hardware.

All classification models are trained over 24 epochs. As there is a strong correlation between valence/arousal and emotion, transfer learning can be exploited to produce the required valence/arousal models more easily. As such, the output layers of the trained emotion classifiers can be removed and replaced with appropriate output layers for the regression task (described below). The resulting models are then fine-tuned over 16 epochs. Both training times were chosen based on those in [25] and resulted in a plateau in validation loss towards the end of training.

B. Architecture 1

This architecture is based on AlexNet [18], including a series of incrementally smaller convolution kernels starting

[2]AffectNet also includes a large number of images automatically annotated by models trained on the manually annotated images.

Fig. 1: Convolution block structures.

(a) Standard. (b) Depth-wise separable.

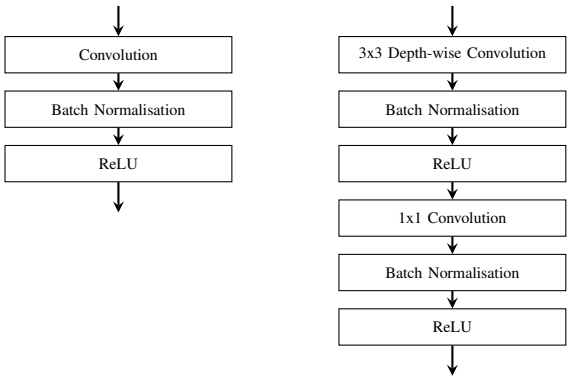


TABLE I: CNN architecture 1, baesd on AlexNet.

Type	Shape	Output
Conv	$9 \times 9 \times 16$	$128 \times 128 \times 16$
MaxPool	2×2	$64 \times 64 \times 16$
Conv	$7 \times 7 \times 32$	$64 \times 64 \times 32$
MaxPool	2×2	$32 \times 32 \times 32$
Conv	$5 \times 5 \times 64$	$32 \times 32 \times 64$
MaxPool	2×2	$16 \times 16 \times 64$
Conv	$3 \times 3 \times 128$	$16 \times 16 \times 128$
MaxPool	2×2	$8 \times 8 \times 128$
Conv	$3 \times 3 \times 128$	$8 \times 8 \times 128$
MaxPool	2×2	$4 \times 4 \times 128$
Flatten	2048	–
2×Dense	1024	–
Dense	8 or 2	1 label or 2 floats

at 9×9 and reducing to 3×3 with 2×2 max-pooling layers in between each convolution block and two fully connected (dense) layers prior to the output layer. There is a 0.2 Gaussian dropout after each pooling layer and a 0.5 dropout after each dense layer.

Unlike the AlexNet architecture, each convolution block is constructed from a conventional 2D convolution layer followed by a batch normalisation layer [15] and a ReLU activation layer [27], as visualised in Fig. 1a, this helps to provide regularisation and speed up training.

The full architecture specification is given in Table I, the output layer contains 8 nodes with soft-max activation for emotion classification and 2 nodes with linear activation for valence/arousal regression.

C. Architecture 2

This architecture is fairly similar to the AlexNet inspired design above, though instead uses the principle behind VGG16 [32] of stacked 3×3 convolution kernels to capture larger image structure. Again the convolution blocks of Fig. 1a are used, interspersed with max-pooling layers and followed by two fully connected layers before the output layer. As above, each pooling layer is followed by a 0.2 Gaussian dropout and there is a 0.5 dropout after each dense layer. The full architecture is given in Table II.

TABLE II: CNN architecture 2, based on VGGNet.

Type	Shape	Output
2×Conv	$3 \times 3 \times 16$	$128 \times 128 \times 16$
MaxPool	2×2	$64 \times 64 \times 16$
2×Conv	$3 \times 3 \times 32$	$64 \times 64 \times 32$
MaxPool	2×2	$32 \times 32 \times 32$
2×Conv	$3 \times 3 \times 64$	$32 \times 32 \times 64$
MaxPool	2×2	$16 \times 16 \times 64$
2×Conv	$3 \times 3 \times 128$	$16 \times 16 \times 128$
MaxPool	2×2	$8 \times 8 \times 128$
2×Conv	$3 \times 3 \times 128$	$8 \times 8 \times 128$
MaxPool	2×2	$4 \times 4 \times 128$
Flatten	2048	–
2×Dense	1024	–
Dense	8 or 2	1 label or 2 floats

TABLE III: CNN architecture 3, based on MobileNet.

Type	Shape	Stride	Output
Conv	$3 \times 3 \times 32$	2	$64 \times 64 \times 32$
DConv	$3 \times 3 \times 64$	1	$64 \times 64 \times 64$
DConv	$3 \times 3 \times 128$	2	$32 \times 32 \times 128$
DConv	$3 \times 3 \times 128$	1	$32 \times 32 \times 128$
DConv	$3 \times 3 \times 256$	2	$16 \times 16 \times 256$
DConv	$3 \times 3 \times 256$	1	$16 \times 16 \times 256$
DConv	$3 \times 3 \times 512$	2	$8 \times 8 \times 512$
5×DConv	$3 \times 3 \times 512$	1	$8 \times 8 \times 512$
DConv	$3 \times 3 \times 1024$	2	$4 \times 4 \times 1024$
DConv	$3 \times 3 \times 1024$	1	$4 \times 4 \times 1024$
GlobalAvePool	1024	–	–
Dense	8 or 2	–	1 label or 2 floats

D. Architecture 3

This architecture is almost identical to that of MobileNet [14], which leverages 3×3 depth-wise separable convolution layers followed by 1×1 conventional convolution layers to retain high performance while minimising architectural complexity. This results in far smaller, tunable, network architectures perfect for deployment to mobile devices.

The design of these depth-wise convolution blocks is shown in Fig. 1b and the full architecture is given Table III. The reduced layer-wise complexity allows for a much deeper model which also retains good width. The output layer remains as above, but no pooling layers are present (stride in convolution layers is instead used for downsampling) other than the final global average pooling layer which replaces the conventional fully connected layers. This pooling layer is followed by a dropout at rate 0.3.

E. Results

All architectures are evaluated on the AffectNet validation set[3] using the metrics provided for the baselines in [25].

For emotion classification accuracy (ACC), F1-score (F1), Cohen's kappa [8] (KAPPA), Krippendorff's alpha [17] (ALPHA), area under precision-recall curve (AUCPR) and area under ROC curve (AUC) are used. For valence/arousal

[3]The AffectNet test set is not yet publicly available.

TABLE IV: Emotion classification performance metrics for each architecture against weighted-loss baseline.

	Baseline	Arch. 1	Arch. 2	Arch. 3
ACC	**0.58**	0.56	**0.58**	0.56
F1	**0.58**	0.56	**0.58**	0.56
KAPPPA	0.51	0.50	**0.52**	0.50
ALPHA	0.51	0.50	**0.52**	0.50
AUCPR	0.56	0.61	**0.62**	0.60
AUC	0.82	**0.90**	**0.90**	0.89

TABLE V: Valence (V) and arousal (A) regression performance metrics for each architecture against weighted-loss baseline.

	Baseline		Arch. 1		Arch. 2		Arch. 3	
	V	A	V	A	V	A	V	A
RMSE	**0.37**	0.41	0.41	0.39	0.41	**0.37**	0.42	0.38
CORR	**0.66**	0.54	0.59	0.53	0.62	**0.56**	0.59	0.53
SAGR	0.74	0.65	0.73	0.74	**0.75**	**0.75**	0.73	0.74
CCC	**0.60**	0.34	0.54	0.43	0.57	**0.48**	0.55	0.47

regression RMSE, Pearson's correlation coefficient (CORR), sign agreement metric [28] (SAGR) and concordance correlation coefficient [20] (CCC) are used.

Emotion classification results are given in Table IV; Arch. 2 outperforms Arch. 1 and 3 in all metrics and outperforms the baseline in all but accuracy and F1 which are equalled at 58%.

Valence/arousal regression results are shown in Table V. As with emotion classification, Arch. 2 gives the best results of the three proposed architectures for both valence and arousal, though only marginally. All proposed architectures perform better for arousal than for valence, also outperforming the baseline. In contrast, the baseline performs significantly better for valence than arousal, also outperforming all proposed architectures.

The increased spatial efficiency of Arch. 3 and consequently greater depth and width do surprisingly little to improve the performance of the model over Arch. 1 and 2, the scaled down versions of more conventional architectures.

All models have a file size close to the goal of 15MB, with the Arch. 2 the largest at 15MB and Arch. 3 the smallest at 13.2MB. All of these remain viable for mobile deployment as described at the start of this section, and have performance close to the baseline for the AffectNet dataset.

The confusion matrix for Arch. 2 (Table VI) shows the classification breakdown for **N**eutral, **H**appy, **Sa**d, **Su**prised, **Af**raid, **D**isgusted, **An**gry and **C**ontemptuous for the validation set containing 500 examples of each emotion. Happiness had the highest rate of correct classifications (72%), while anger had the lowest with just 43% correct, often being confused with disgust.

III. MOBILE APPLICATION

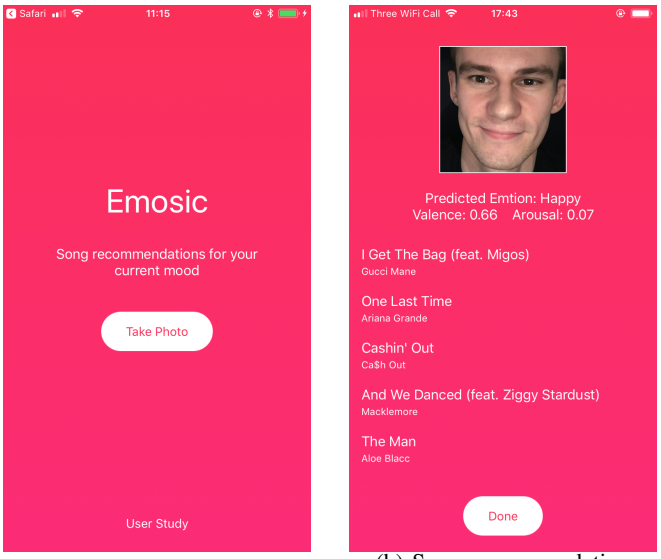
A. Interface

The mobile application is implemented in Swift for the iOS platform and has a very simple interface as shown in Fig. 2. The user opts to take a photo, which prompts the native camera interface to be presented using the front-facing camera. Once the user has taken a photo of their face, the emotion, valence and arousal are predicted and the results presented on the screen in Fig. 2b. In normal operation the predicted emotion, valence and arousal are shown to the user along with the top five recommended songs, clicking on each song opens it in the Spotify app for the user to listen to.

TABLE VI: Arch. 2 emotion classification confusion matrix for the AffectNet validation set.

	N	H	Sa	Su	Af	D	An	C
N	247	7	52	60	11	22	34	67
H	20	358	6	26	4	15	4	67
Sa	63	9	279	22	38	41	37	11
Su	33	22	15	298	97	20	7	8
Af	21	6	32	72	320	32	12	5
D	29	9	36	24	31	316	42	13
An	71	4	39	22	29	98	216	21
C	71	56	12	21	3	33	26	278

Fig. 2: Emosic app user interface.



(a) Main screen.

(b) Song recommendation screen.

B. Affect Recognition

In order to be used for prediction within the iOS app, the highest performing Keras models described in Sec. II (Arch. 2) are converted for use with Apple's CoreML framework [2]. Apple provides an open source Python toolkit, Coremltools [4], for this purpose. Almost no additional modification is required for the models to function within the iOS app, only minor preprocessing of the input image data.

To match the input format described in Sec. II-A, Apple's Vision framework [3] is used to determine the bounding box for the user's face, this is then cropped and resized to the required 128×128 pixels. The image data can then be fed directly to the applicabale model after conversion to pixel buffer format.

C. Song Recommendation

Song recommendations are obtained using the Spotify Web API recommendations service [34]. This service provides a REST endpoint which can be given up to five seed genres, a modality (major or minor) and numeric values for valence and energy (taken to be analogous to arousal) between 0 and 1. A list of songs which best match the inputs are then returned in JSON format.

Seed genres are determined using a predefined mapping from the emotion predicted by the CoreML classifer (one of the basic eight) to a list of five seed genres. Predicted valence and arousal from the CoreML regression model are used directly after translation from the output range of $[-1, 1]$ to the required $[0, 1]$. The desired modality is taken from the sign of the valence prediction, positive being major and negative minor.

D. User Study

The user study is built directly into the app and can be accessed from the bottom of the screen in Fig. 2a. The user is first given instructions regarding the study and information about data retention, they are then presented with the screens shown in Fig. 3 in order from left to right.

Firstly, the user is told an emotion which they are to emulate (Fig 3a), they then take a photo using the native camera interface. The recommended songs are presented with a 5-star rating input (Fig. 3b) and finally a self-annotation screen for valence and arousal (Fig. 3c). This sequence of screens is repeated for ten distinct emotions: neutral, delighted, happy, miserable, sad, surprised, angry, afraid, disgusted and contemptuous. These are chosen to correspond closely with the eight classified emotions as well as providing some variance in valence and arousal.

The study was completed 17 times by female and male subjects aged between 21 and 55. The results are broken down as in Sec. II-E, with classification predictions compared against instructed emotions, and valence/arousal predictions compared against self-annotated values. The inexperience of subjects in self-annotation has likely led to discrepancies in what is taken to be ground-truth for the study (particularly for arousal which some users struggled to understand), though the results obtained should still provide useful insight as to the effectiveness of the application as a whole.

The predictions made by the deployed models in the user study are evaluated using the same metrics as for the AffectNet dataset described in Sec. II-E, the results are shown in Tables VIIa and VIIb. The confusion matrix for the user study is given in Table VIII; for classification performance the results for delighted and miserable are used to represent annotated emotions happy and sad in order to obtain a balanced set which is directly comparable to the results in Sec. II-E.

Emotion classification results are slightly worse than for the AffectNet validation set and vary greatly between emotions. Happy has an accuracy of 100% while contempt is at just 6%, often being confused with sadness. Surprise is often misclassified as fear and neutral as contempt. Valence/arousal prediction is much more successful and outperforms the AffectNet results in almost all metrics. Valence prediction is notably improved, with a RMSE of 0.28 and a CCC score of 0.81, while arousal remains much closer to the AffectNet scores.

TABLE VII: User study results.

(a) Emotion classification.

ACC	0.54
F1	0.50
KAPPPA	0.47
ALPHA	0.47
AUCPR	0.71
AUC	0.93

(b) Valence/Arousal prediction.

	Valence	Arousal
RMSE	0.28	0.32
CORR	0.83	0.51
SAGR	0.75	0.76
CCC	0.81	0.50

TABLE VIII: User study emotion classification confusion matrix.

	N	H	Sa	Su	Af	D	An	C
N	7	1	3	1	0	0	0	5
H	0	17	0	0	0	0	0	0
Sa	3	0	13	0	1	0	0	0
Su	0	0	0	5	12	0	0	0
Af	0	0	0	3	14	0	0	0
D	0	0	1	0	4	11	1	0
An	1	1	3	0	2	4	5	1
C	2	1	7	0	2	3	1	1

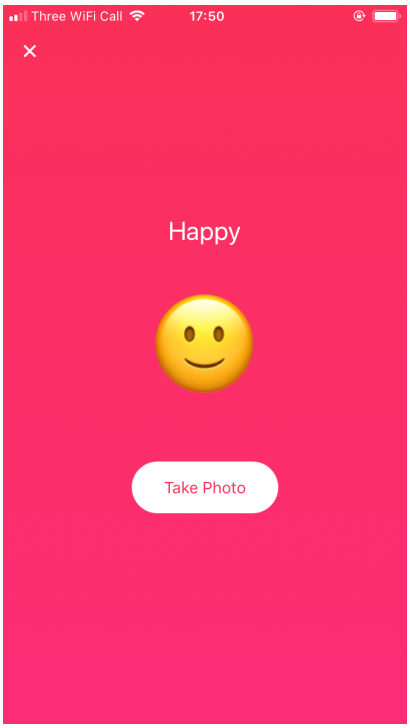
The average rating for song recommendations was 3.32 indicating that users had a generally positive view of the application's functionality, emotion specific ratings are shown in Table IX. Emotions which might be viewed to have clearer musical connotations (e.g. happy, sad) had higher ratings than those with perhaps less clear connotations (e.g. contempt, disgust). The average rating for correct classifications was 3.53 and for incorrect classifications was 3.01.

For this application a reduced number of more pertinent emotions would likely have proved more effective, along with a training set optimised for photos featuring the user's face from the front only, rather than the varied poses present in AffectNet.

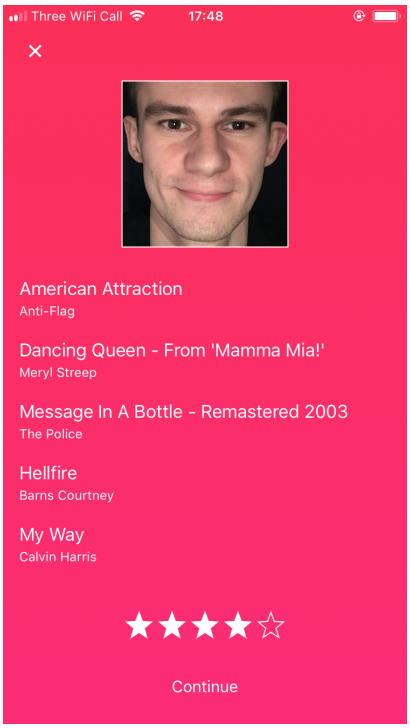
TABLE IX: User study song recommendation ratings.

Neutral	3.59
Delighted	3.59
Happy	3.71
Miserable	3.59
Sad	3.59
Surprised	3.70
Angry	3.35
Afraid	2.65
Disgusted	2.76
Contemptuous	2.65
Total	3.32

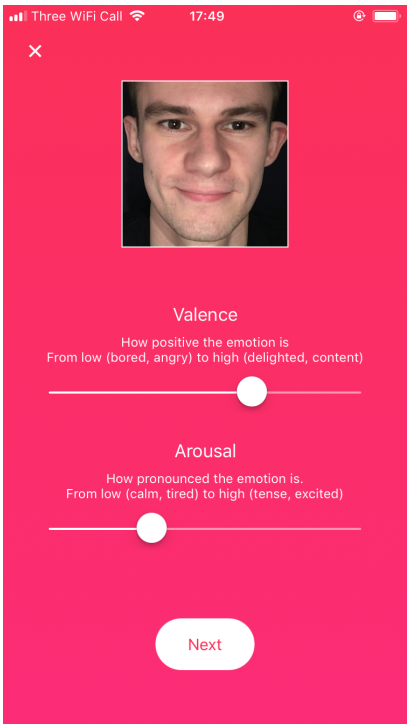
Fig. 3: User study interface.



(a) Screen instructing the user which emotion to emulate.



(b) Song recommendation screen with rating input.



(c) Affect self annotation screen with valence and arousal sliders.

IV. CONCLUSIONS AND FUTURE WORKS

A. Conclusions

In this paper three CNN architectures for affect recognition are present with the aim of minimising storage requirements for mobile deployment. These models gave comparable results to the current baseline when evaluated on the AffectNet dataset [25]. The trained models were deployed in a music recommendation app and a user study performed to assess their real world performance. The results of the user study showed the models to perform similarily when deployed compared with AffectNet, and that users were generally happy with the application's functionality. These results support the proposition that EIUIs are an area of great potential within affective computing and are now becoming increasingly feasible in a real-world setting.

B. Future Works

The functionality of the Emosic application could easily be integrated into a fully-featured music application such as Spotify and expanded to great effect. For example, determining a user's affect each time they manually choose a song would allow a model to be built up over time to provide tailored predictions for that user.

In general a user is unlikely to be as expressive as they are prompted to be in this study which may reduce the effectiveness. The recent rise of wearables may provides a solution to this, as a number of modalities useful for affective computing, such as heartbeat, are now more readily available within mobile applications. These could easily be incorporated into multi-modal models along with accelerometer or usage activity data to improve the accuracy and reduce invasiveness of emotion recognition in a mobile setting.

It is not difficult to see how this kind of emotionally intelligent behaviour could be expanded to many other application domains, though privacy concerns are of course an issue with this sort of activity. Emphasis will need to be placed on clearly explaining what systems like this will be doing to users, and keeping computation local with as little long-term data retention as possible.

To facilitate widescale adoption of EIUI continued research into very efficient (both in terms of file-size and computationally) deep neural network architectures will be required; Google's MobileNets [14] are a very promising start in this respect. Alternative structures such as Inception may provide a more efficient basis than the options presented in this paper and lower floating point precision could be an easy way to cut model size, though the impact on performance may be significant.

It is likely that major developments will need to be driven by popular smartphone manufacturers at an OS level, as is already beginning to happen with Apple's CoreML [2] and Vision frameworks [3]. This will allow for larger, more complex and likely more accurate models to be trained for common tasks such as emotion classification as many of the constraints discussed would no longer apply.

V. ACKNOWLEDGMENTS

Thanks to Turner Stone Ltd. for provision of the NVIDIA GeForce GTX 1080 Ti and associated hardware used for the training and evaluation of deep learning models.

REFERENCES

[1] I. Abdić et al. Driver frustration detection from audio and video in the wild. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, IJCAI'16, pages 1354–1360. AAAI Press, 2016.

[2] Apple Inc. CoreML Framework. https://developer.apple.com/documentation/coreml, 2017.

[3] Apple Inc. Vision Framework. https://developer.apple.com/documentation/vision, 2017.

[4] Apple Inc. and contributors. CoreML Community Tools. https://github.com/apple/coremltools, June 2017.

[5] O. Arriaga, M. Valdenegro-Toro, and P. Plöger. Real-time convolutional neural networks for emotion and gender classification. *CoRR*, abs/1710.07557, 2017.

[6] F. Benitez-Quiroz, R. Srinivasan, and A. M Martinez. Emotionet: An accurate, real-time algorithm for the automatic annotation of a million facial expressions in the wild. *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 06 2016.

[7] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[8] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

[9] D. Duncan, G. Shine, and C. English. Facial emotion recognition in real time. Stanford, 2017.

[10] I. J. Goodfellow et al. Challenges in representation learning: A report on three machine learning contests. *Neural Networks*, 64(Supplement C):59 – 63, 2015. Special Issue on "Deep Learning of Representations".

[11] Google Inc. and contributors. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[13] C. Hewitt. Emosic. https://github.com/friggog/Emosic, January 2018.

[14] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.

[15] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015.

[16] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[17] K. Krippendorff. Estimating the reliability, systematic error and random error of interval data. *Educational and Psychological Measurement*, 30(1):61–70, 1970.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[19] G. Levi and T. Hassner. Emotion recognition in the wild via convolutional neural networks and mapped binary patterns. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, ICMI '15, pages 503–510, New York, NY, USA, 2015. ACM.

[20] L. I.-K. Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, 45(1):255–268, 1989.

[21] G. K. Mark, D. Alan, and A. Jen. Affective videogames and modes of affective gaming: Assist me, challenge me, emote me. In *Proceedings of the 2005 DiGRA International Conference: Changing Views: Worlds in Play*, 2005.

[22] D. McDuff, R. el Kaliouby, T. Senechal, M. Amr, J. F. Cohn, and R. Picard. Affectiva-mit facial expression dataset (am-fed): Naturalistic and spontaneous facial expressions collected "in-the-wild". In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, CVPRW '13, pages 881–888, Washington, DC, USA, 2013. IEEE Computer Society.

[23] D. Messinger, L. L. Duvivier, Z. Warren, M. Mahoor, J. Baker, A. S. Warlaumont, and P. Ruvolo. Affective computing, emotional development, and autism. In *The Oxford Handbook of Affective Computing*. Oxford Library of Psychology, 2014.

[24] Microsoft Corporation. Microsoft Emotion API. https://azure.microsoft.com/en-gb/services/cognitive-services/emotion/, 2017.

[25] A. Mollahosseini, B. Hasani, and M. H. Mahoor. Affectnet: A database for facial expression, valence, and arousal computing in the wild. *IEEE Transactions on Affective Computing*, PP(99):1–1, 2017.

[26] A. Mollahosseini, B. Hassani, M. J. Salvador, H. Abdollahi, D. Chan, and M. H. Mahoor. Facial expression recognition from world wild web. *CoRR*, abs/1605.03639, 2016.

[27] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pages 807–814, USA, 2010. Omnipress.

[28] M. A. Nicolaou, H. Gunes, and M. Pantic. Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space. *IEEE Transactions on Affective Computing*, 2(2):92–105, 2011.

[29] O. M. Parkhi, A. Vedaldi, A. Zisserman, et al. Deep face recognition. In *BMVC*, volume 1, page 6, 2015.

[30] E. Politou, E. Alepis, and C. Patsakis. A survey on mobile affective computing. *Computer Science Review*, 25(Supplement C):79 – 100, August 2017.

[31] J. Schwan, E. Ghaleb, E. Hortal, and S. Asteriadis. High-performance and lightweight real-time deep face emotion recognition. *SMAP*, pages 76–79, 07 2017.

[32] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[33] A. Singh, N. Bianchi-Berthouze, and A. C. Williams. Supporting everyday function in chronic pain using wearable technology. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 3903–3915, New York, NY, USA, 2017. ACM.

[34] Spotify AB. Spotify Web API. https://developer.spotify.com/web-api/.

[35] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[36] Thwis Inc. CoreML Store. https://coreml.store, 2017.