Emotion Detection using Audio and Video Detection

Chaitanya Goswami
cgoswami

Rajnish Aggarwal
rajnisha

Nithila Yalamanchi
nyalaman

Abstract

Inspired by recent advances in multi-task learning, we want to propose an algorithm for training a classifier for emotion recognition which can leverage information from both audio and visual data. As humans tend to express emotions through both audio and visual cues, combining these cues could greatly enhance the capabilities of an emotion classifier.

1 Problem Definition

We are given data from $\mathcal{K} = 2$ domains, let the kth data set be represented by $\{x_i^k, y_i^k\}_{i=1}^{n_k}$, where x_i^k is the raw data for emotion classification in the kth domain, and y_i^k denotes the class label corresponding to a specific emotion. We want to jointly learn a classifier leveraging data from both domains.

2 Background

Our basic approach rested on the assumption that there exists a common latent space into which data from both modalities (audio and image) can be projected into and learning a joint classifier in this domain should yield us a better classification. To do so, we wanted to jointly learn a classifier $\mathcal{C}_{\text{Joint}}$ and two mappings $\phi_a(.), \phi_i(.)$ jointly which map the data from audio and image domain to the common latent space respectively. To this effect we formulated a loss function which looked like:

$$\mathcal{L} = \mathcal{L}_{\mathcal{C}}(\phi_a(x_{audio})) + \alpha\mathcal{L}_{\mathcal{C}}(x_{image}) + \beta\text{Reg}(\mathcal{C}_{\text{Joint}}) + \gamma\text{Similarity}(\phi_a(x_{audio}), \phi_i(x_{image}))$$

.

where $\mathcal{L}_{\mathcal{C}}(\phi_a(x_{audio}))$ signifies the loss incurred by the joint classifier on audio data projected into the latent space, $\mathcal{L}_{\mathcal{C}}(\phi_v(x_{image}))$ is the loss incurred by the joint classifier on into data projected into the latent space, $\text{Reg}(\mathcal{C}_{\text{Joint}})$ is the regularizer on the joint classifier and $\text{Similarity}(\phi_a(x_{audio}), \phi_i(x_{image}))$ is a similarity constraint which forces the output of two mappings to be similar. We tried preliminary experiments with binary classification but since the dataset becomes too small because we were only considering two classes ,we couldn't get coherent results out of it.

3 Related Work And Dataset

So to get a sense of how emotion classification works we first looked at how emotion classification works in both audio and video domains individually. And also we wanted to look at ways of how to combine these features which is inspired by multi-task learning approaches. For this we see the review paper of [2].

For the audio part, [3] and [4] provide ways to extract the LLD as well as higher level features, while [8] provides a deep way of extracting features. For the image part, [6] and [7] provide us with an idea

of the features typically used in facial emotion recognition models. [9] provides a full end to end deep architecture for solving the multi-modal emotion recognition problem. [12] and [13] compare more example architectures that could be used for handling multi-modal features. We were also quite intrigued by the attributes learning approach mentioned typically employed by zero-shot learning mentioned in [10].

Right now we settled on the dataset RAVDESS [1] which contains recordings of 24 professional actors, vocalizing lexically-matched statements in a neutral North American accent. The different emotions includes calm, happy, sad, angry, fearful, surprise, and disgust expressions, so 8 classes.We are considering this dataset because it contains both face-and-voice. There are 7536 recordings which have the face image as well as the corresponding audio to it, so for a given event we have corresponding data in both audio and video domain, and each recording is labelled with a particular emotion. Also they have versions in which they have song and speech recordings

4 Methods/Models Developed

The core idea that we had, was that the for a task such as Emotion Classification, there should exist a joint latent space, into which we can project our data from both modalities (video and audio), and ideally should be able to leverage information from both domains to do better classification rather than just using one particular modality. To that effect, we tried five different architectures:

4.1 Pre-processing Of the Data

We preprocessed our data in the following manner, for audio we took both the song and speech data, then for each of these recordings we sampled them using "Kaiser best" sampling and took the first 40 MFCCs for each audio recording.

For image we took the videos from song and every half second we took two frames, which came about to roughly 6 frames every video. We concatenate these two frames horizontally as shown in figure 1. We did this because experimentally we found out that adding that redundancy helps the network figure out easily the important features from the image.

Now since the number of images were higher than the number of audio recordings we just copied the recordings to ensure one to one mapping between audio and video data-set which we feed into our joint architectures. (We did this class wise, so that for each class we have the same number of audio recordings and images)

Figure 1: Image Example

4.2 Individual Classifiers For Baseline Comparisons

We implemented two CNN architectures, that is an "Audio" CNN which classifies emotion on just the audio features, and an "Image" CNN which classifies based on Image features. These individual classifiers serve as baselines against which we compare joint architectures described below.

4.3 Architecture 1: Simple Concatenation

This is the most basic architecture that we implemented, which is shown in Fig 2. This architecture consists of two CNN networks: a "Audio" CNN and an "Image" CNN. The "Audio" CNN is just

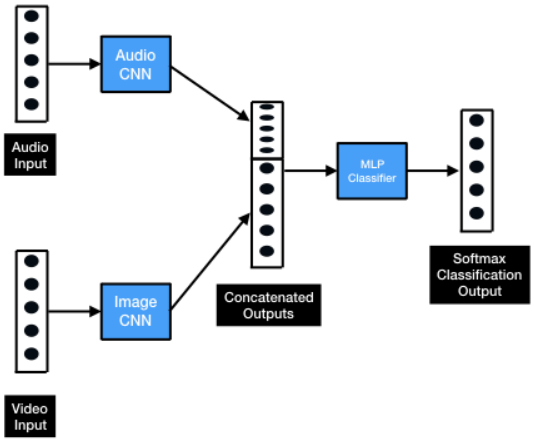

Figure 2: Simple Concatenation

a two layer CNN with a single Maxpool layer, whose final output is flattened to get the joint latent representation of the audio features. Similarly the "Image" CNN is also a two-layer CNN with maxpool whose output is again flattened to get the representation of Image features in the joint latent space(One thing to note is that the output of the both the "Image" and "Audio" CNN is not of the same dimension). Now we concatenate the output of the above two CNNs to get the joint representation of the data and feed it into a three layer MLP classifier whose loss is just the cross-entropy loss. While providing input to this data we make sure that both audio and the video feature that is being fed into the architecture to create this concatenated vector belong to the same class, i.e. if audio being fed to the network belongs to the class "Happy", then the corresponding Image being fed also belongs to the class "Happy".

The basic intuition behind such an architecture was that the respective CNNs will try to learn the mappings into the common latent space for both audio and image. After we project both audio and video into the common space, since video and audio should hopefully contain different cues/information about the emotion, concatenating them will help the joint classifier classify the emotion better than just giving the classifier the data from only one modality.

4.4 Architecture 2: Simple Concatenation with LSTM

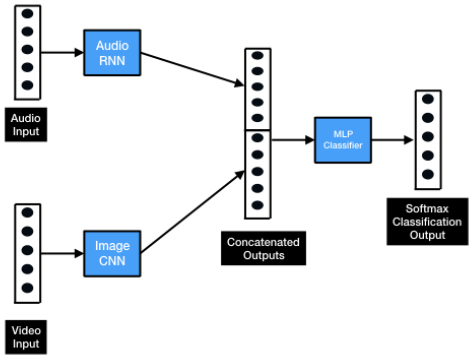

Figure 3: Simple Concatenation with LSTM

Now this architecture is the same as architecture 1 but instaed of an Audio "CNN" we used an "Audio" LSTM and tried to see how it would affect the performance.

The intuition behind using an LSTM was that since audio has a temporal structure exploiting that might give us a better latent representation for audio features but our experiments showed similar performance with CNNs, so we didn't experiment with LSTMs since they make model much larger without adding any significant advantage.

4.5 Architecture 3: Simple Concatenation with Dimension Equalization

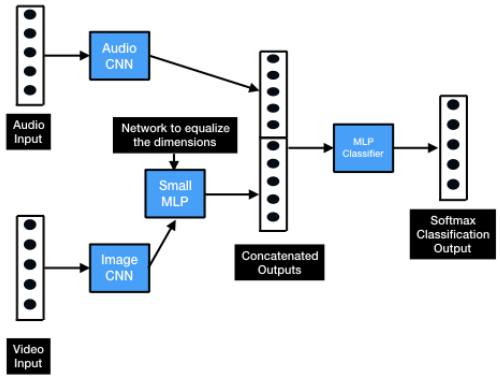

Figure 4: Simple Concatenation with Dimension Equalization

We modified architecture 1 by adding a small MLP network so that the output of both Audio "CNN" and Video "CNN" (which is being concatenated) is the same in dimension. We did this to try alleviate the effect of one modality dominating the other one. Such a phenomenon can happen since the output of the image CNN has much higher dimension than the output of "Audio" CNN which can lead to the image modality dominating the network and ignoring the contribution of the "Audio" part. As a result "Audio" CNNs might not be learnt properly.

4.6 Architecture 4: Probe Network

This architecture is also similar to architecture 3 but we added two other "probe" networks which essentially play the role of regularizer. These probe networks are essentially MLP classifier which classify on both audio and video data separately. The input for the "Audio" probe network is the output of "Audio" CNN and the input to the "Image" probe network is the output of the "Image" CNN after the dimension equalization. Both network also try to classify the emotion based on their respective input and have a cross entropy loss. The loss for this whole architecture is defined as sum of the loss of the joint classifier and the sum of losses of both probe networks.

$$\mathcal{L}_{\text{Architecture}} = \mathcal{L}_{\text{Joint Classifier}} + \mathcal{L}_{\text{Image Probe}} + \mathcal{L}_{\text{Audio Probe}}$$

Where each individual loss is a cross entropy loss. The basic intuition behind this kind of architecture was that during training the whole architecture might favor one modality over the other and tries to learn based only on one modality(which is typically the one that gives the best accuracy individually), so to prevent that we also added these two network which try to ensure that while learning the joint representation, also ensure that the representation also work individually.

4.7 Architecture 0: Joint Loss Architecture

This was the initial architecture proposed in the midway report. This is very similar to architecture 3 but the only difference is that there is no concatenation of the outputs of the "Audio" and "Image" CNNs rather we directly feed the output of each CNN network into the joint MLP classifier network. To explain how the loss is calculated. Let x_i^{audio}, x_j^{image}, having labels(one-hot encoded) y_i^{audio}, y_j^{image}, let N_{audio}, N_{image} be the total number of training points in "Audio" and "Image" dataset and β is a hyperparameter which decides how much importance to give each modality, then

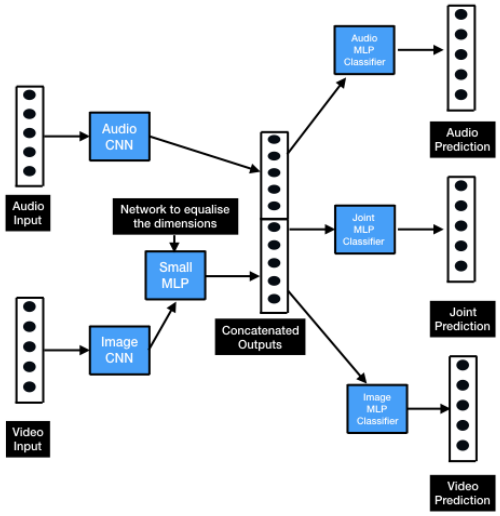

Figure 5: Probe Network

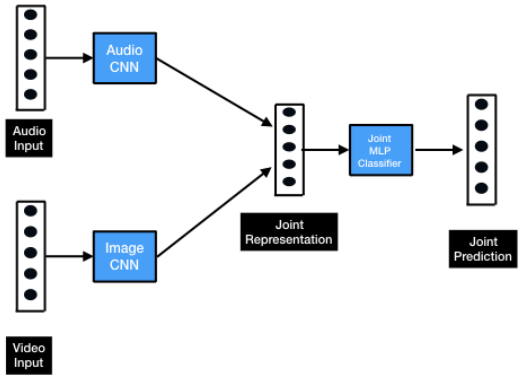

Figure 6: Joint Loss Network

(it is basically cross entropy loss)

$$\mathcal{L}_{architecture} = \sum_{i=1}^{N_{audio}} \sum_{l=1}^{K} y_i^{audio} \log(\text{JointMLP}(\text{AudioCNN}(x_i^{audio})))$$

$$+\beta \sum_{i=1}^{N_{Image}} \sum_{l=1}^{K} y_j^{image} \log(\text{JointMLP}(\text{ImageCNN}(x_i^{image})))$$

. The idea behind this network was to transform the audio and video into a common space but not by concatenating rather by means of the loss function but this model is prone to overfitting, so we didn't explore it further.

5 Experiments and Results

5.1 Experiment 1

The goal of this experiment was to see how our joint architectures compared against their individual counterparts.

5.1.1 Experimental Setup

We took the data described in section 4.1, and took a 80-10-10 split, with 80% of the data being used for training and 10% for test and validation separately. We trained the individual CNN classifier for both "Audio" and "Image" data separately to serve as a baseline, and then we trained all the five architecture mentioned in section 4, and compared their best test accuracies with respect to their individual counterparts. All the networks were trained using SGD optimizer with momentum and for 25 epochs. We considered all classes available in the dataset for this classification (which were 8).

5.1.2 Results

Network type	Test Accuracy	Validation Accuracy	Train Accuracy
Audio CNN	72%	72%	99%
Image CNN	85%	84%	98%
Architecture 1	**90%**	91%	99%
Architecture 2	**89%**	89%	98%
Architecture 3	**89%**	90%	99%
Architecture 4	**91%**	91%	98%
Architecture 0	78%	79%	99%

5.1.3 Observations

As you can see that every joint architecture except architecture 0 gives substantial improvement over individual classifiers (best accurcay 85%), hence it confirms our intuition that using data from both modalities helps improve classification of emotion.

As you can see architecture 0 is prone to overfitting, as can be seen from the gap in train and test accuracies, which suggests introduction of regularization.

Architecture 2 is just replacing the CNN in architecture 1 by an LSTM. Since adding an LSTM does not give us any additional gain we don't consider Architecture 2 in our further experiments.

5.2 Experiment 2

From the previous experiment we inferred that architectures 1-4 all performed almost similarly on the dataset, so to compare between these different architectures, we tested how there performances are affected when we decrease the amount of data we have, so as to see which architectures tries to learn the most robust representation.

5.2.1 Experimental Setup

To see the effect of small amounts of data we carried out the following steps. First for this experiment we only consider architecture 1, 3 and 4. To simulate the effect of small amounts of data we change thedata split from data 80-10-10 to two cases 50-50, and 25-75, where 50% and 25% of the total dataset is considered for training and rest is used for test. We don't tune any hyperparameters for this experiment, rather we use the hyperparameter used in experiment 1. We again use SGD optimizer with momentum and train for 25 epochs. We calculate train and test accuracies.

5.2.2 Results

50% Train Data

Network Type	Test Accuracy	Train Accuracy
Architecture 1	84%	99%
Architecture 3	85%	99%
Architecture 4	**86%**	98%

25% Train Data

Network Type	Test Accuracy	Train Accuracy
Architecture 1	73%	99%
Architecture 3	78%	99%
Architecture 4	**80%**	98%

5.2.3 Observations

With 50% data we still see that the performance is similar across architectures although architecture 4 tends to outperform the other architectures, but as we decrease the amount of training data to even smaller amounts we start to see that architecture 4 performs considerably better than the others.

5.3 Experiment 3

Based on experiment 2, we see that architecture 4 tends to be a little bit better since it is able to learn a robust representation as compared to the other ones, so we tried to plot the validation accuracy for this model.

5.3.1 Experimental Setup

We trained architecture 4 for 25 epochs with SGD optimizer with momentum and plotted the validation accuracy for the 80-10-10 split. We plotted validation accuracy for the joint classifier as well as both video and audio probe networks. Since we had probe networks here we could calculate the individual accuracies as well, which was not possible for the other architectures.

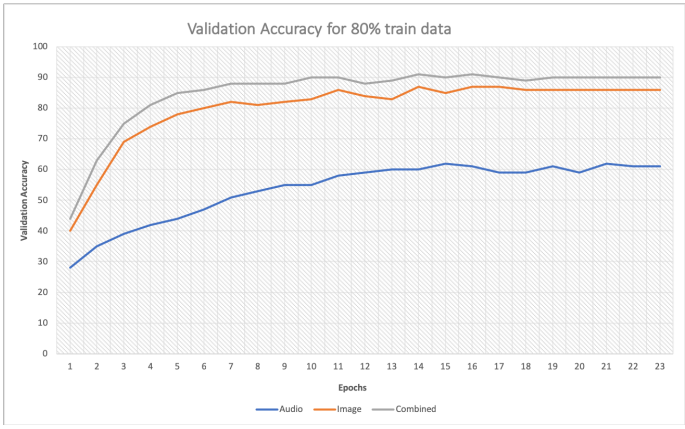

Figure 7: Validation Accuracies For architecture 4

6 Discussion and Explanation Of Results

Our experiments gave us a lot of insights on how to make cross-modal architectures. Experiment 1 essentially confirmed our belief that combining data from both modalities is actually helpful, and that projecting them into a common latent space is feasible way to do this. Also one more conclusion that we can make from experiment 1 is that when you have enough data the architecture of the network does not matter much, as almost all our joint architectures gave similar performances. Since all the architectures try to project data into a common space and then classify it, with enough data all of them are able to do it. One another thing of note is that architecture 0 does not do well at all on the joint classification, which provided us with another insight that directly trying to transform both audio and video features into a common representation as compared to concatenated one doesn't work. If you look again at the accuracy of architecture 0 it is sort of an average of accuracies of both individual classifiers, which seems to suggest that joint classifier in order to balance the loss terms of both audio and video ends up learning a representation that is kind of in between. Hence it shows that such an approach is not a good idea as compared to concatenation because as opposed to concatenation in which the joint classifier can look at different cues from both video and audio

simultaneously, in this case the architecture ends up learning a representation which is a mash of both modalities and hence ends up doing worse than just an individual classifier. The last thing we see that since architecture 1 and architecture 2 have similar accuracies, so adding an LSTM does not give any significant improvements but we attribute this to the fact that the dataset is quite "clean" and maybe with noisier dataset one still might have to consider LSTMs instead of CNNs to exploit the temporal structure. But for the sake of this project we excluded LSTMs from experiments since they take a lot of time to train and didn't seem to add any significant value to the joiny classifier

With Experiment 2 we tried to compare the three architectures which were doing well on the dataset. Now the three networks differed in terms of regularization, and we wanted to see this effect. Now in architecture 1, the output of "Audio" and "Image" CNN has different dimensions(with the size of output of "Image" CNN is too high compared to "Audio CNN"), so it is very easy in this architecture for the network to bias itself towards the image modality since the it forms the dominant part of the vector that is being fed into the MLP network. With enough data the network ultimately figures out the audio part also but our intuition pointed us to the fact that with less data this architecture would start to perform considerably worse. So to tackle this problem we added a small MLP network in front of the "Image" CNN to equalize the dimensions. We thought that this should alleviate the problem partially, but this still doesn't address the problem at its core, since the audio and video features have an inherent asymmetry to them(as can be seen from their individual classification accuracy). Audio typically tends to perform way worse than images, hence to address this problem we made architecture 4 which is the probe network architecture. The idea behind the probe network is that it tries to prevent the scenario in which one modality dominates the other one. Essentially if the image modality tries to dominates then the the loss of audio probe network would start becoming higher and hence the gradients respect to the audio probe loss would be dominant and therefore in backpropagation it would force all the networks to try to learn representation for audio which performs better on audio individually hence counterbalancing the effect of "image" modality and vice-versa. So according to us, this has to have the best regularization effect, and this is what we observe in experiment 2, as we decrease data architecture 1 suffers the most. Architecture 3 does better than architecture 1 but architecture 4 outperforms all of them in the low data regime, which we believe is due to the regularization of its probe networks.

The plot of validation accuracy for architecture 4 also gives insight into how the network behaves during training as you can see the joint classifier always remains above the individual classifier which is a desirable behaviour but also you can see that that all the three classifiers saturate almost around similar epochs which further strengthen our argument that the probe networks does not allow one modality to dominate because otherwise one of the modality (typically image in this case) would converge faster and then audio(if converges) would converge at a very later stage and typically at a point worse than its individual accuracy which is not the case in probe architecture.

7 Future Work

As we see that architecture 4 seems to have a good regularization effect. We had the idea to extend that thought by distillation. So essentially instead of training the probe networks on cross-entropy loss we use logits of the individual classifiers (trained seperately) and try to match the outputs of the probe network to these logits. This helps in two ways since we can reduce model complexity of the architecture and also we can use priors induced by huge networks by distilling them into these small probe networks and hence hopefully regularizing them better. Another dimension could be exploring architecture 0, since due to the nature of its architecture we can do augmented learning for audio features. By this we mean, by using image features we can learn a better representation for audio features and then maybe do better classification based on these features, but since the architecture is a little prone to overfitting we still need to come up with a way to regularize it. The reason we believe this would work is because the image typically has better representational power and by training them jointly in this fashion we can maybe extract more relevant features out of the audio.

Acknowledgments

We want to acknowledge Prof. Ruslan Salakhutdinov, Brynn Edmunds and all the TAs of the course 10707 for their continued support and for providing us with this opportunity to do something exciting as part of our course project.

References

[1] Livingstone, Steven R., and Frank A. Russo. "The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English." PloS one 13.5 (2018): e0196391.

[2] Zhang, Yu, and Qiang Yang. "A survey on multi-task learning." arXiv preprint arXiv:1707.08114 (2017).

[3] Anagnostopoulos, Christos-Nikolaos, Theodoros Iliou, and Ioannis Giannoukos. "Features and classifiers for emotion recognition from speech: a survey from 2000 to 2011." Artificial Intelligence Review 43.2 (2015): 155-177.

[4]Zhang, Biqiao, Emily Mower Provost, and Georg Essl. "Cross-corpus acoustic emotion recognition from singing and speaking: A multi-task learning approach." ICASSP. 2016.

[5]Schuller, Björn, et al. "The INTERSPEECH 2013 computational paralinguistics challenge: social signals, conflict, emotion, autism." Proceedings INTERSPEECH 2013, 14th Annual Conference of the International Speech Communication Association, Lyon, France. 2013.

[6]Mehta, Dhwani, Mohammad Faridul Haque Siddiqui, and Ahmad Y. Javaid. "Facial Emotion Recognition: A Survey and Real-World User Experiences in Mixed Reality." Sensors 18.2 (2018): 416.

[7]Xiaoxi, Ma, et al. "Facial emotion recognition." Signal and Image Processing (ICSIP), 2017 IEEE 2nd International Conference on. IEEE, 2017.

[8]Cummins, Nicholas, et al. "An Image-based deep spectrum feature representation for the recognition of emotional speech." Proceedings of the 2017 ACM on Multimedia Conference. ACM, 2017.

[9]Tzirakis, Panagiotis, et al. "End-to-end multimodal emotion recognition using deep neural networks." IEEE Journal of Selected Topics in Signal Processing 11.8 (2017): 1301-1309.

[10]Fu, Yanwei, et al. "Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content." IEEE Signal Processing Magazine 35.1 (2018): 112-125.

[11]Han, Jing, et al. "From Hard to Soft: Towards more Human-like Emotion Recognition by Modeling the Perception Uncertainty." Proceedings of MM '17. ACM, 2017.

[12]Zheng, Ziqi, et al. "Multimodal Emotion Recognition for One-Minute-Gradual Emotion Challenge." arXiv preprint arXiv:1805.01060 (2018).

[13]Pini, Stefano, et al. "Modeling Multimodal Cues in a Deep Learning-based Framework for Emotion Recognition in the Wild." Proceedings of ICMI '17. ACM, 2017.

[14]Williams, Jennifer, et al."Recognizing Emotions in Video Using Multimodal DNN Feature Fusion." Proceedings of the First Grand Challenge and Workshop on Human Multimodal Language (Challenge-HML). ACL, 2018.

[15] Barros, Pablo, and Stefan Wermter. Developing crossmodal expression recognition based on a deep neural model. Adaptive Behavior 24(5): 373-396. 2016.

[16]Arriaga, Octavio, Matias Valdenegro-Toro, and Paul Plöger. "Real-time Convolutional Neural Networks for Emotion and Gender Classification." arXiv preprint arXiv:1710.07557 (2017).