A comparison of shared encoders for multimodal emotion recognition

Sai Anuroop Kesanapalli, Riya Ranjan, Aashi Goyal, Wilson Tan
[kesanapa, riyaranj, aashiarv, wtan1167] @usc.edu
University of Southern California

1 Problem definition

Multimodal learning aims to create models that process and relate information from multiple modalities. Human communication is multimodal by nature which limits the performance of unimodal models. In this work, a shared encoder architecture that is capable of fusing multimodal information while providing better synergy between modalities is compared to architectures that use separate encoders.

To this end, we developed unimodal audio, unimodal video, and a multimodal pipeline that builds on the former. We employ various classes of shared encoders such as 2D CNNs comprising ResNet18 (7), GoogLeNet (15), and VGG16 (14); 3D CNNs comprising Simple3D CNN, and I3D (4); and 2D Vision Transformer (ViT) (6) and 3D Vision Transformer (VideoMAE) (16). We test our pipelines on the task of emotion recogntion on full-scale version of CREMA-D dataset (3) that contains 7442 videos of actors expressing 6 kinds of emotions in various intensities.

We present a principled comparison of the performance of different pipelines and encoders, identify the achievements and shortcomings of these architectures, and discuss the implications along with the possibilities for future work.

2 Literature Review

(2) provides architectures that have one encoder tailored per modality. These are specific to voice assistants on smart-watches that utilize accelerometer readings and audio cues. We wish to use a common encoder rather than independent ones. (10) leverages the benefits of complementary information provided by different types of labels and develop three ranking models based on SVM, DNN, and GBDT. This direction is orthogonal to our approach, yet an interesting one to consider since their task is emotion recognition as well. (11) proposes one sensor fusion model that is designed for Radar and Lidar data, both of which are vision in nature. Moreover they employ a student-teacher framework. Despite the differences, our work draws inspiration from their sensor fusion pipeline, albeit customized for audio-vision data in our case. (18) proposes a method where normalization parameters are exchanged between modes for implicit feature alignment. However they too employ one encoder per modality. Previous works have also leveraged attention mechanisms for fusion. (5) presents a simple modality-agnostic model by using self and cross attention on images and text to learn a common embedding space. Using transformer architectures which utilizes attention mechanisms may also be beneficial for our audio-vision task. (12) proposes HighMMT, an architecture scalable with modalities. Our pipelines share structural similarities with HighMMT, albeit we employ multiple classes of shared encoders, such as 2D CNN, 3D CNN, and Transformer, rather than devising a customized Transformer-based architecture.

3 Data Description

We utilize the Crowd Sourced Emotional Multimodal Actors Dataset (CREMA-D) (3) for our work, offering a rich multimodal experience, integrating audio and video for enhanced emotion analysis. Evaluated by over $2,400$ individuals, CREMA-D includes $7,442$ video clips with performances by 91 actors, providing a diverse exploration of emotional expression. Within the dataset, each actor presents 12 sentences, expressing 6 emotions at different intensity levels. Each video clip is brief, lasting less than 5 seconds. Importantly, the dataset includes the number of ratings for each emotion, offering valuable insights into the perceived emotional content of the performances. Note: there are 3 videos with recording issues which gives a total of 7439 good videos.

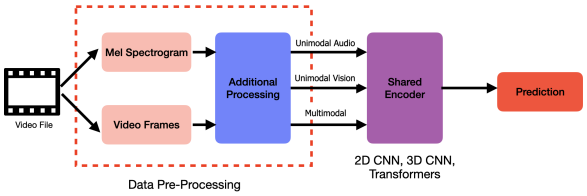


Figure 1: Architecture

4 Method

Our architecture is visualized in Fig. 1. Videos are pre-processed (Section 5) to generate frames containing faces, Mel spectrograms, concatenated if the pipeline is multimodal, along with additional processing depending on the architectural requirements of the encoder, then passed-on to the encoder which performs emotion recognition. The type of concatenation, number of channels used in the processed images, additional processing such as generation of patches for ViT and temporal alignment for 3D CNN and VideoMAE, vary depending upon the encoder employed. This architectural design draws inspiration from the plug-and-play ideology, with shared encoder being the changeable component.

5 Experiments & Results

5.1 2D Encoders

5.1.1 2D CNNs

ResNet18 (7), VGG16 (14), and GoogLeNet (15) are employed in this class of encoders. These CNNs were chosen with regards to their number of learnable parameters – GoogLeNet ($\sim$ 7M), ResNet18 ($\sim$ 11.7M), VGG16 ($\sim$ 138M), which provide us with a wide spectrum. This choice also reflects the need for consideration into deployment of this class of architectures on edge-devices in the Internet of Things (IoT) era.

5.1.2 ViT

A simple ViT (6) is employed in this experiment. This transformer accepts 1 input channel, has a patch dimension of 16, dimensionality of token embeddings is 768, has $6/8$ transformer blocks, $4/8$ attention heads, dimensionality of linear layer is 1024, and includes an additional classification token. This transformer has $\sim$ 16.4M learnable parameters (with 6 transformer blocks and 4 attention heads).

5.2 2D Experiments

For video preprocessing, frames were extracted from videos and resized to 224×224 (Width $\times$ Height) images. Of these, the middle frame was chosen to perform face-detection using a Multi-Task Cascaded Convolutional Neural Network (MTCNN) (8), and the frame was then cropped to the detected face. For audio preprocessing, audio `WAV` files were extracted out of the video `FLV` files, and Mel spectrograms were generated using `librosa` at a sample rate of $22,050$ Hz, 2048 FFT points, hop length of 512, and 512 Mel bands. These spectrograms are then resized to 224×224 images. For some 2D CNNs in the unimodal and multimodal pipelines, images (both faces and spectrograms) are converted to grayscale whereas for the rest of the 2D CNNs and ViT pipelines, they remain multi-channeled. In the multimodal pipeline, these faces and spectrogram images are concatenated horizontally to form a single chunk of multimodal data which is the passed-on to the encoder employed in the pipeline (2D CNN, ViT) which do further processing on this as per their architectural requirements. Experiments have been conducted on a full-scale version of the CREMA-D datset, with a $70/30$ train-test split that corresponds to 5209 train samples, 2233 test samples, and 50 epochs. We note here that these include the 3 samples that had inherent recording errors.

5.3 2D Results

5.3.1 Unimodal Audio

Results are described in Table 1. We observe that GoogLeNet performs better in comparison to ResNet18, VGG16, and ViT in terms of test accuracy. This is evident from the barplot in Figure 2a. We attribute this observation to quality of pre-trained weights of VGG16 atop which we fine-tune the model on Mel spectrograms. We believe ViT performs a lot worse when compared to CNNs because of the patching scheme we employed which is not ideal in the case of spectrograms.

5.3.2 Unimodal Vision

Results are described in Table 2. We observe that VGG16 performs better in comparison to ViT, ResNet18, and GoogLeNet in terms of test accuracy. This can also bee seen from the barplot in Figure 2b.

5.3.3 Multimodal

Results are described in Table 3. We observe that here too GoogLeNet outperforms other 2D encoders, as can be seen from the barplot in Figure 2c. Interestingly, although VGG16 performs better than the rest in vision modality, it is not the case in multimodal regime. Moreover, when compared to unimodal case, test accuracies have improved for ResNet18 and GoogLeNet in the multimodal scenario. However, VGG16 is seen to have taken a hit in the accuracy in the multimodal scenario when compared to the unimodal cases.

5.4 2D Discussion

2D CNNs: The classic CNNs – ResNet18 (7), VGG16 (14), and GoogLeNet (15) perform decently on the test split, with GoogLeNet outperforming others in unimodal audio and multimodal scenarios, and VGG16 doing the best in case of unimodal vision. As expected, train accuracies of these encoders are higher than their test accuracies, indicating that these encoders are not generalizing that well. Moreover it is observed that the difference in train and test accuracies varies considerably across encoders. Surprisingly, VGG16 has a higher test accuracy than train in the multimodal scenario. Of the three modalities, audio accuracies are considerably lower than the rest. This is probably due to two reasons – much information regarding emotion of the speaker is not contained in the audio when compared to vision, and Mel spectrogram conversion may be leading to loss of some information.

ViT: Contrary to our initial guess, ViT does not always perform better than 2D CNNs. In the audio modality, it performs much worse when compared to vision and multimodal scenarios. This is attributed to patching scheme as mentioned earlier. Even in case of unimodal vision and multimodal scenarios, its performance is almost as a good as the corresponding top performing 2D CNN. An explanation for this lies in the observation that ViTs are known to outperform CNNs, but only when trained on large datasets (14-300M images), as mentioned in (6). In our case, the entire dataset consists of 7442 video clips, and correspondingly those many video frames as per our pre-processing scheme. For training, as mentioned earlier in this section, this number is 5209. However, we do note here that despite this stark difference in the number of samples required to train, ViT does perform comparably to 2D CNNs, owing to its superior architecture involving attention mechanisms.

General Discussion: We tried different sets of hyperparameters (Table 4) for each class of 2D encoders and modalities, such as batch size, learning rate, dropout rate (in case of ViT), and reported the best values obtained. We fixed training epochs to 50, a convenient choice with regards to execution time, and one that also corresponds to a point beyond which the test accuracy does not improve further.

5.5 3D Encoders

We also look at 3D CNNs and Video Transformers as shared encoders:

Simple3D CNN: A simple CNN that uses 6 convolutional layers followed by a final classifcation layer. The purpose of this model is to serve as a baseline for 3D performance.

I3D: (4) Uses a series of inception modules, where each inception module is made up of several 3D convolutional layers, followed by average pooling over spatial and temporal dimensions to make a prediction. This model is a one-stream RGB version pretrained on ImageNet.

VideoMAE: (17) A masked autoencoder (MAE) that extends to videos by using the vanilla ViT as a backbone. It does this by masking random 3D patches in videos as opposed to 2D patches found in 2D MAEs. This model was pretrained on the Kinetics dataset (9).

5.6 3D Experiments

Due to recording errors in the dataset, 3 videos were removed resulting in a total of 7439 total videos. The 3D experiments use the full 7439 videos and a randomly selected 80/20 train-test split which gives a total of 5951 training and 1488 testing samples.

For video preprocessing, frames were extracted from videos at 24 frames per second and resized to 224×224 (Width $\times$ Height) images. For audio preprocessing, mel spectrograms were created using audio files then converted to 3D. To convert to 3D, the mel spectrograms were evenly divided into chunks along the time-axis. The number of chunks they were divided into varied to match the number of frames extracted from their corresponding video. This was done to temporally align frames

with spectrogram chunks. Mel spectrogram chunks were then resized to 224×224 images. Frames and spectrogram chunks retained RGB color channels. In the end, we are left with arrays with dimensions # of frames/chunks, color channels, width, height, i.e., (# of frames/chunks, 3, 244, 244). Now that frames and spectrogram chunks are temporally aligned, they were concatenated together widthwise to form the 3D multimodal data.

However, there are still two issues. Firstly, dimensionality mismatch in the video transformer as they are commonly unable to handle rectangular data. This is easily resolved by further resizing video frames to 208×224 and spectrogram chunks to 16×224 before concatenation. Secondly, an issue of varying # of frames/chunks per array. This is handled differently depending on architectures:

- **3D CNNs:** Padding was used to resolve the issue by concatenating blank images until all arrays had the same # of frames/chunks as the longest array (135).
- **VideoMAE:** Instead of padding, 32 frames/chunks were taken evenly spread across the # of frames/chunks to maintain good temporal fidelity while substantially lowering memory usage.

The final data dimensions look as follows:

CNNs Unimodal: (135, 3, 244, 244)
CNNs Multimodal: (135, 3, 488, 244)
VideoMAE Unimodal: (32, 3, 244, 244)
VideoMAE Multimodal: (32, 3, 244, 244)

5.7 3D Results

In addition to comparing 3D models against each other, we will also compare them to human performance. (3) provides human performance along with the CREMA-D dataset for audio-only, vision-only, and audio-vision emotion classification at 40.9%, 58.2% and 63.6% respectively.

5.7.1 Unimodal Audio

Both 3D CNN models outperformed humans on unimodal audio emotion classification. Out of the two models, I3D performed best as shown in Table 1. However, Simple3D is by far the smaller model with only 3262 parameters compared to I3D with 12.3M parameters.

VideoMAE performs worse than Simple3D and humans, but better than random guessing. This is likely due to spatial redundancy which will be elaborated in the 3D discussion section.

5.7.2 Unimodal Vision

Although Simple3D was unable to outperform humans in unimodal vision, I3D significantly does. It is likely that the model was able to transfer learn from ImageNet pretraining to boost unimodal vision performance.

VideoMAE was unable to do better than random guessing performance. More on this in the 3D discussion section.

5.7.3 Multimodal

Simple3D again is unable to outperform humans in this case, however, it was able use multimodal interaction to get a higher test accuracy compared to any of its unimodal versions. I3D still performs well and better than humans, but it does not do better than its best unimodal variant (vision), which could mean that classification is largely skewed by vision. One simple way to check is to look at model performance with modality ablation. These results are represented as Ablated I3D. Results showed a significant drop in performance when either vision 1 or when audio is removed. This suggests that both modalities are important for I3D multimodal classification (i.e., likely no overly dominant modality).

As for VideoMAE, because the multimodal results are similar to the unimodal audio, and because it was randomly guessing on unimodal vision, it is likely that audio is the completely dominant modality.

5.7.4 3D Discussion

Video Transformers: Although video transformers show good results on other datasets like Kinetics, they struggle with spatial redundancy (13) which Kinetics mitigates with diverse actions and environments (9). Spatial redundancy is inherently an issue with videos, but it is especially challenging for facial emotion recognition where facial action units may persist for the duration of the emotional state which adds to spatial redundancy. In the frequency domain, this is mitigated slightly, but still not enough to give good predictions. Furthermore, joint-space attention used in VideoMAE also scales quadtratically with respect to both image size and number of frames (1). Adding a small 3D CNN model may help mitigate both issues.

A 3D CNN can be used to recognize important

temporal and spatial features. Not only would this shrink image sizes and number of frames through convolution and pooling, this may also solve the spatial redundancy. Although adding a 3D CNN adds to memory usage which is counter-intuitive to saving memory, judging from the results in Table 3, a small 3D CNN like Simple3D with only 3K parameters can already provide decent features.

3D CNNs: They good. Simple3D tiny model but decent already.

Converting Audio to 3D: Might be a waste of weights, but it does help with multimodal interaction. Also worked well for small 3D CNN model which gave better results than some large 2D models.

General Discussion: Hyperparameter tuning. With hyperparameter tuning, probably much better results.

5.8 2D vs 3D

I3D outperforms all encoders across all modalities, except for GoogLeNet in the case of unimodal audio. A superior architecture composed of a series of Inception modules (Section 5) is the key to its performance. However, we observe from Tables 1, 2, and 3, it is not the norm that 3D encoders work better than their 2D counterparts. This is evident, for example, from the performance of VGG16 which is better than Simple3D CNN and Ablated I3D for vision modality, and GoogLeNet in case of audio modality and multimodal data. This may be a crucial factor while weighing-in the pros and cons of using 2D vs 3D encoders, where the latter, although give a better accuracy, are costlier to train in terms of compute resources and time. 2D pre-processing is comparatively cheaper computation-wise when compared to 3D pre-processing, which is a determining factor in making a choice between 2D and 3D encoders. Considering training time as a proxy, ResNet18 takes about 8.33 minutes to train on multimodal data for 50 epochs on an NVIDIA Tesla P100 GPU accessed through USC CARC. Whereas, a 3D encoder like I3D takes about 45 minutes per epoch to train on CARC. In scenarios where these models are to be deployed on the edge, 2D encoders, CNNs especially, have an upper-hand thus. However, where compute is not a constraint, and for mission-critical applications with low tolerance for misclassifications, 3D encoders are an ideal choice.

6 Conclusion & Future Work

We have successfully implemented the unimodal and multimodal audio and vision pipelines with 2D CNN, 3D CNN, ViT, and VideoMAE as encoders. Moreover, we tested our pipelines on a fullscale version of CREMA-D containing 7442 samples, and all the 6 emotion classes. For all the pipelines and encoders, we tried different combinations of hyper-parameters (in a non-exhaustive manner), and identified the best modes of operation. We compared these pipelines against each other in terms of their test accuracies, reasoned the observed behavior, and analyzed the implications.

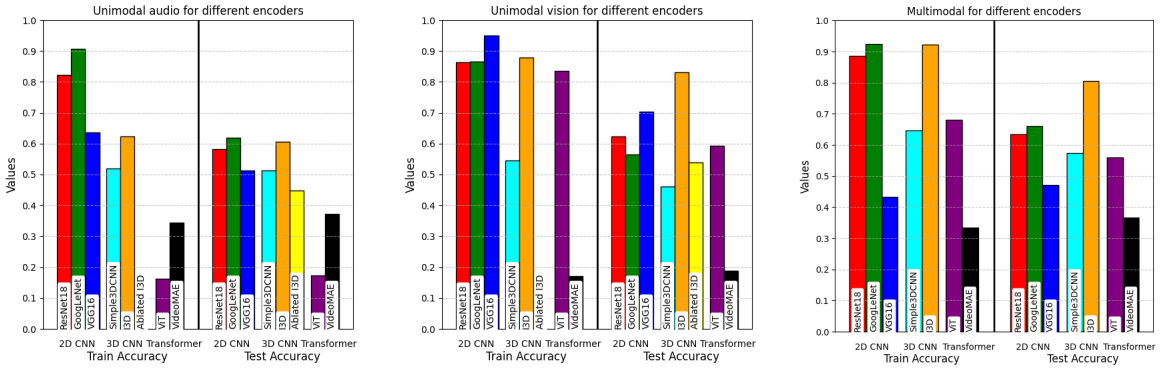
For future work, ViT architecture can be further improved and trained on a much bigger dataset to match the current state-of-the-art performance. Patching of audio modality information encoded as Mel spectrograms is not really an ideal choice. A better thing to do is to replicate these spectrograms across the patches and concatenate these replicated spectrograms with the patched video frames. This, we believe, will improve the performance of our pipelines with ViT significantly. In the 3D pipeline, adding a small 3D CNN may help mitigate spatial redundancy in videos and also address joint-space attention used in VideoMAE that scales quadratically with respect to both image size and number of frames. Finally, each experiment can be run multiple times and the averaged metrics of these set of experiments along with error bars can be reported, as a better practice.

7 Miscellaenous

Code-base hosted on GitHub (private) repository - https://github.com/ksanu1998/multimodal_course_project. Our experiments are available as `.ipynb` notebooks and `.py` scripts accompanied with README files and can be reproduced. Please contact any of the team members for access and information. I3D PyTorch implementation was taken from https://github.com/piergiaj/pytorch-i3d/tree/master. ViT PyTorch implementation was adapted from https://theaisummer.com/vision-transformer/.

8 Contributions

- Anuroop



(a) Unimodal audio (b) Unimodal vision (c) Multimodal

Figure 2: Comparison of accuracies of encoders for different modalities

Type	Encoder	Train Loss	Train Acc.	Test Loss	Test Acc.
2D CNN	ResNet18	1.06	0.8216	1.04	0.5825
	GoogLeNet	1.137	0.907	1.4209	0.619
	VGG16	1.4038	0.6366	1.5259	0.5120
3D CNN	Simple3D CNN	1.213	0.519	1.294	0.514
	I3D	0.991	0.623	1.022	0.605
	Ablated I3D	-	-	2.707	0.448
Transformer	ViT	1.7931	0.1634	1.7908	0.1738
	VideoMAE	1.566	0.344	1.512	0.372

Table 1: Unimodal audio metrics for different encoders

Type	Encoder	Train Loss	Train Acc.	Test Loss	Test Acc.
2D CNN	ResNet18	1.0687	0.8634	1.3649	0.6225
	GoogLeNet	1.179	0.866	1.470	0.566
	VGG16	1.0952	0.9495	1.3337	0.7040
3D CNN	Simple3D CNN	1.160	0.546	1.417	0.462
	I3D	0.334	0.878	0.463	0.831
	Ablated I3D	-	-	1.928	0.540
Transformer	ViT	0.4378	0.8361	1.6823	0.5934
	VideoMAE	1.795	0.170	1.790	0.188

Table 2: Unimodal vision metrics for different encoders

Type	Encoder	Train Loss	Train Acc.	Test Loss	Test Acc.
2D CNN	ResNet18	1.1288	0.8854	1.4075	0.6350
	GoogLeNet	1.121	0.925	1.382	0.661
	VGG16	1.6099	0.4329	1.5713	0.4716
3D CNN	Simple3D CNN	0.918	0.647	1.169	0.573
	I3D	0.211	0.923	0.629	0.806
Transformer	ViT	0.8365	0.6811	1.3221	0.5598
	VideoMAE	1.575	0.334	1.513	0.366

Table 3: Multimodal metrics for different encoders

Class	Encoder	# (Train + test)	bs	lr	opt	Loss	ep	dr	GPU
3D CNN	Simple3D(V)	5951 + 1488	8	0.001	Adam	CE	27	0	A100 40GB
	Simple3D(A)						36		
	Simple3D(M)								
	I3D(A)						5	0.5	
	I3D(V)						9		
	I3D(M)		4				16		
2D CNN	ResNet18(A)	5209 + 2233	32	0.0001			50	-	P100
	ResNet18(V)			0.001					
	ResNet18(M)								
	GoogLeNet(A)			0.0001					
	GoogLeNet(V)		16						
	GoogLeNet(M)		32						
	VGG16(A)			0.001					
	VGG16(V)			0.00001					
	VGG16(M)			0.001					
Transformer	ViT(A)		16	0.0001				0.4	T4
	ViT(V)		32					0.2	P100
	ViT(M)		16					0.4	T4
	VideoMAE(A)	5951 + 1488	8	0.001			3	0.5	A100 80GB
	VideoMAE(V)						4		
	VideoMAE(M)								

Table 4: Training setup

1. Responsible for implementing unimodal audio and vision, multimodal pipelines with 2D CNN - ResNet18
2. Midterm presentation deck and report
3. Responsible for implementing unimodal audio and vision, multimodal pipelines with ViT, and for running the corresponding full-scale experiments for unimodal audio and multimodal pipelines
4. Responsible for 2D data pre-processing for full-scale experiments
5. Midterm and Final presentation deck and report – including barplots, analysis, and discussion on 2D experiments

- Riya
  1. Unimodal audio and vision pipelines with 2D CNN - GoogLeNet
  2. Multimodal pipeline with 2D CNN - GoogLeNet
  3. Fine tune GoogLeNet model on batch size and learning rates, with best fit results, analysis and discussions
  4. Testing out ViT fullscale experiments with different combinations of batch size, heads, blocks, dropout rates and analysis.
  5. Midterm and Final report
- Aashi
  1. Unimodal audio and vision pipelines with 2D CNN - VGG16
  2. Mulitomodal pipeline with 2D CNN - VGG16
  3. Fine-Tuning of VGG16 model, with best fit results, analysis and discussions
  4. Testing out ViT fullscale experiments with different combinations of batch size and learning rates and analysis.
  5. Midterm and Final report
- Wilson
  1. Solely responsible for all of 3D parts including: 3D data preprocessing, 3D encoder training/testing (unimodal and multimodal), 3D analysis and discussion, etc.
  2. Midterm and Final report

All team members have actively contributed to the project. Furthermore, everyone contributed to proof-reading both the presentation deck and the report.

References

[1] BERTASIUS, G., WANG, H., AND TORRESANI, L. Is space-time attention all you need for video understanding?, 2021.

[2] BUDDI, S. S., SARAWGI, U. O., HEERAMUN, T., SAWNHEY, K., YANOSIK, E., RATHINAM, S., AND ADYA, S. Efficient multimodal neural networks for trigger-less voice assistants, 2023.

[3] CAO, H., COOPER, D. G., KEUTMANN, M. K., GUR, R. C., NENKOVA, A., AND VERMA, R. Crema-d: Crowd-sourced emotional multimodal actors dataset. *IEEE transactions on affective computing 5*, 4 (2014), 377–390.

[4] CARREIRA, J., AND ZISSERMAN, A. Quo vadis, action recognition? a new model and the kinetics dataset, 2018.

[5] DODDS, E., CULPEPPER, J., HERDADE, S., ZHANG, Y., AND BOAKYE, K. Modality-agnostic attention fusion for visual search with text feedback, 2020.

[6] 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. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations* (2021).

[7] HE, K., ZHANG, X., REN, S., AND SUN, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[8] JIANG, B., REN, Q., DAI, F., XIONG, J., YANG, J., AND GUI, G. Multi-task cascaded convolutional neural networks for real-time dynamic face recognition method. In *Communications, Signal Processing, and Systems* (Singapore, 2020), Q. Liang, X. Liu, Z. Na, W. Wang, J. Mu, and B. Zhang, Eds., Springer Singapore, pp. 59–66.

[9] KAY, W., CARREIRA, J., SIMONYAN, K., ZHANG, B., HILLIER, C., VIJAYANARASIMHAN, S., VIOLA, F., GREEN, T., BACK, T., NATSEV, P., SULEYMAN, M., AND ZISSERMAN, A. The kinetics human action video dataset, 2017.

[10] LEI, Y., AND CAO, H. Audio-visual emotion recognition with preference learning based on intended and multi-modal perceived labels. *IEEE Transactions on Affective Computing 14*, 4 (2023), 2954–2969.

[11] LI, Y.-J., PARK, J., O'TOOLE, M., AND KITANI, K. Modality-agnostic learning for radar-lidar fusion in vehicle detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2022), pp. 918–927.

[12] LIANG, P. P., LYU, Y., FAN, X., TSAW, J., LIU, Y., MO, S., YOGATAMA, D., MORENCY, L.-P., AND SALAKHUTDINOV, R. High-modality multimodal transformer: Quantifying modality & interaction heterogeneity for high-modality representation learning, 2023.

[13] SELVA, J., JOHANSEN, A. S., ESCALERA, S., NASROLLAHI, K., MOESLUND, T. B., AND CLAPES, A. Video transformers: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023), 1–20.

[14] SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[15] SZEGEDY, C., LIU, W., JIA, Y., SERMANET, P., REED, S., ANGUELOV, D., ERHAN, D., VANHOUCKE, V., AND RABINOVICH, A. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015), pp. 1–9.

[16] TONG, Z., SONG, Y., WANG, J., AND WANG, L. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems* (2022), S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35, Curran Associates, Inc., pp. 10078–10093.

[17] TONG, Z., SONG, Y., WANG, J., AND WANG, L. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training, 2022.

[18] YIN, Y., XU, J., ZU, T., AND SOLEYMANI, M. X-norm: Exchanging normalization parameters for bimodal fusion. In *Proceedings of the 2022 International Conference on Multimodal Interaction* (2022), pp. 605–614.