# MFAS: Multimodal Fusion Architecture Search

Juan-Manuel Pérez-Rúa[1,3*] Valentin Vielzeuf[1,2*]
Stéphane Pateux[1] Moez Baccouche[1] Frederic Jurie[2]

[1]Orange Labs, Cesson-Sévigné, France
[2]Université Caen Normandie, France
[3]Samsung AI Centre, Cambridge, UK

## Abstract

*We tackle the problem of finding good architectures for multimodal classification problems. We propose a novel and generic search space that spans a large number of possible fusion architectures. In order to find an optimal architecture for a given dataset in the proposed search space, we leverage an efficient sequential model-based exploration approach that is tailored for the problem. We demonstrate the value of posing multimodal fusion as a neural architecture search problem by extensive experimentation on a toy dataset and two other real multimodal datasets. We discover fusion architectures that exhibit state-of-the-art performance for problems with different domain and dataset size, including the* NTU RGB+D *dataset, the largest multimodal action recognition dataset available.*

## 1. Introduction

Deep neural networks have demonstrated to be effective models for solving a large variety of problems in several domains, including image [22] and video [5] classification, speech recognition [15], and machine translation [44], to name a few. In a multimodal setting, it is very common to transfer models trained on the individual modalities and merge them at a single point. It can be at the deepest layers, known in the literature as *late fusion*, which is relatively successful on a number of multimodal tasks [40]. However, fusing modalities at their respective deepest features is not necessarily the most optimal way to solve a given multimodal problem. We argue in this paper that considering features extracted from all the hidden layers of independent modalities could potentially increase performance with respect to only using a single combination of late (or early) features. Thus, this work tackles the problem of finding good ways to combine multimodal features to better exploit the information embedded at different layers in deep learning models for classification.

Our hypothesis is in line with a common interpretation of deep neural models considering that features learned in a convolutional neural network carry varying levels of semantic meanings. In vision, for example, lower layers are known to serve as edge detectors with different orientations and extent, while further layers capture more complex information such as semantic concepts, like *faces, trees, animals*, etc. Evidently, it is difficult to determine by hand what is the most optimal way of mixing features with varying levels of semantic meaning when solving for multimodal classification problems. For example, learning to classify *furry animals* might require analysis of lower level visual features that can be used to build up the concept of fur, whereas classes like *chirping birds* or *growling* might require analysis of more complex audiovisual attributes. Indeed, features from different layers at different modalities can give different insights from the input data. A similar idea is exploited by unimodal ResNets [14], where features from different depths are utilized by later layers through skip connections.

In this line of thought, a few recent works analyzed other possible combinations from input modalities [38, 42]. However, those methods fall short, as the model designer needs to choose empirically which intermediate features to consider. Evaluating all of the possibilities by hand would be extremely intensive or simply intractable. Indeed, the more modalities and the deeper they are, the more complicated it is to choose a mixture. This is all the more true when enabling nested combinations of multimodal features. It is in fact a large combinatorial problem.

In order to handle this issue, the aforementioned combinatorial problem has to be tackled by an efficient search method. Luckily, the underlying structure of this problem makes it specially amenable to sequential search algorithms. We propose in this paper to rely on a sequential model-based optimization (SMBO) [19] scheme, which

*Assert joint first authorship. This work was done while JMPR was with Orange Labs.

has previously been applied to the related problem of neural architecture search or *AutoML* [26, 34]. In a few words, we tackle the problem of multimodal classification by directly posing the problem as a combinatorial search. To the best of our knowledge, this is a completely new approach to the multimodal fusion problem, which, as shown by thorough experimentation, improves the state-of-the-art on several multimodal classification datasets.

This paper brings four main contributions: i) an empirical evidence of the importance of searching for optimal multimodal feature fusion on a synthetic toy database. ii) The definition of a search space adapted to multimodal fusion problems, which is a superset of modern fusion approaches. iii) An adaptation of an automatic search approach for accurate fusion of deep modalities on the defined search space. iv) Three automatically-found state-of-the-art fusion architectures for different known and well studied multimodal problems encompassing five types of modalities.

The rest of this paper is organized as follows. In Section 2, we describe the work that is related to ours, including multimodal fusion for classification and neural architecture search. Next, in Section 3 we explain our search space and methodology. In Section 4, we present an experimental validation of our approach. Finally, in Section 5, we give final comments and conclusions.

## 2. Related work

Current design strategies of neural architectures for general classification (multimodal or not) and other learning problems consider the importance of the information encoded at various layers along a deep neural network. Indeed, advances in image classification like *residual nets* [14] and *densely connected nets* [18] are related to this idea. Similarly, for the problem of pose estimation, *stacked hourglass networks* [32] connect encoder and decoder parts of an autoencoder by short-circuit convolutions, allowing the final classifiers to ponder features from bottom layers. However, it is commonly accepted that manually-designed architectures are not necessarily optimally solving the task [48]. In fact, looking at the type of neural networks that are automatically designed by search algorithms, it seems that convoluted architectures with many cross-layer connections and different convolutional operations are preferred [9, 48].

Interestingly, Escorcia *et al*. argued that the visual attributes learned by a neural network are distributed across the entire neural network [11]. Similarly, it is commonly understood that neural networks encode features in a hierarchical manner, starting from low-level to higher-level features as one goes deeper along them. These ideas motivate well our take on the problem of multimodal classification. This is, trying to establish an optimal way to connect and fuse multimodal features. To the best of our knowledge, this work is the first one to directly tackle multimodal fusion for classification as an architecture search problem.

In the following, we give an overview of the multimodal fusion problem for classification as a whole. We then continue with a short discussion on relevant methods for architecture search, since it appears at the core of our method.

**Multimodal fusion.** To categorize the different recent approaches of deep multimodal fusion, we can define two main paths of research: architectures and constraints.

The first path focuses on building best possible fusion *architectures e.g.* by finding at which depths the unimodal layers should be fused. Early works distinguished early and late fusion methods [4], respectively fusing low-level features and prediction-level features. As reported by [40], late fusion performs slightly better in many cases, but for others, it is largely outperformed by the early fusion. Late fusion is often defined by the combination of the final scores of each unimodal branch. This combination can be a simple [39] or weighted [29] score average, a bilinear product [8], or a more robust one such as rank minimization [46]. Thus, methods such as multiple kernel learning [6] and super-kernel learning [43] may be seen as examples of late fusion. Closer to early fusion, Zhou *et al*. [47] propose to use a Multiple Discriminant Analysis on concatenated features, while Neverova *et al* [31] apply a heuristic consisting of fusing similar modalities earlier than the others. Recently, to take advantage of both low-level and high-level features, Yang *et al*. [45] leverage boosting for fusion across all layers. To avoid overfitting due to large number of parameters in multilayer approches, multimodal regularization methods [1, 13, 21] are also investigated. Another architecture approach for multimodal fusion could be grouped under the idea of attention mechanisms, which decides how to ponder different modalities by contextual information. The mixture of experts by [20] can be viewed as a first work in this direction. The authors proposed a gated model that picks an expert network for a given input. As an extension, Arevalo *et al*. [3], proposed Gated Multimodal Units, allowing to apply this fusion strategy anywhere in the model and not only at prediction-level. In the same spirit, multimodal attention can also be integrated to temporal approaches [16, 27].

The second category of multimodal fusion methods proposes to define *constraints* in order to control the relationship between unimodal features and/or the structure of the weights. Ngiam *et al*. [33], proposed a bimodal autoencoder, forcing the hidden shared representation to be able to reconstruct both modalities, even in the absence of one of them. Andrew *et al*. [2], adapted Canonical Correlation Analysis to deep neural networks, maximizing correlation between representations. Shahroudy *et al.* [38], use cascading factorization layers to find shared representations be-

tween modalities and isolate modality-specific information. To ensure similarity between unimodal features, Engilberge *et al.* [10] minimize their cosine distance. Structural constraints can also be applied on the very weights of the neural networks. In addition to modality dropping, Neverova *et al.* [30] propose to zero-mask the cross-modal blocks of the weight matrix in early stages of training. Extending the idea of modality dropping, Li *et al.* [25], propose to learn a stochastic mask. Another structure constraint as done through tensor factorization was proposed by [8].

**Neural architecture search.** The last couple of years have seen an increased interest on *AutoML* methods [9, 26, 34, 35, 48]. Most of these methods rely somehow on a neural module at the core of their respective search approaches. This is now known in the literature as *neural architecture search* (NAS). Neural-based or not, *AutoML* methods were traditionally reserved for expensive hardware configurations with hundreds of available GPUs [26, 48].

Very recently, progressive exploration approaches and weight-sharing schemes have allowed to tremendously reduce the necessary computing power to effectively perform architecture search on sizeable datasets. Another advantage of progressive search methods [26, 34] is that they leverage the intrinsic structure of the search space, by sequentially increasing the complexity of sampled architectures. In this paper, we start from a sequential method with weight sharing [34] and adapt it to the problem of multimodal classification. In particular, we design a search space that is prone to sequential search and which is a superset of previously introduced fusion schemes, *e.g.*, [42]. This is an important aspect of our contribution. As demonstrated by [49], constraining the search space is a key element for affordable architecture search. It turns out that directly tackling multimodal datasets by automatic architecture search without designing a constrained, but meaningful, search space would not be tractable. We demonstrate the value of our approach and the importance of optimizing neural architectures for multimodal classification tasks by tackling three challenging datasets.

## 3. Methodology

In this work, as in many others addressing multimodal fusion, we start from the assumption of having an off-the-shelf multi-layer feature extractor for each one of the involved modalities. In practice, this means that we start from a multi-layer neural network for each modality, which we assume to be already pre-trained. However, the reader should consider that our fusion approach is in fact not limited to neural networks as primary feature extractors.

Without loss of conceptual generality, we assume from now on that we will deal with two modalities. The multimodal dataset is composed by pairs of input and output data

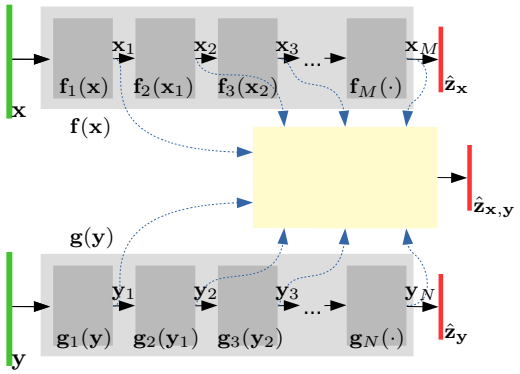


Figure 1. **General structure of a bi-modal fusion network**. Top: A neural network with several hidden layers (grey boxes) with input $\mathbf{x}$, and output $\hat{\mathbf{z}}_\mathbf{x}$. Bottom: A second network with input $\mathbf{y}$, and output $\hat{\mathbf{z}}_\mathbf{y}$. In this work we focus on finding efficient fusion schemes (yellow box and dotted lines).

$(\mathbf{x}, \mathbf{y}; \mathbf{z})$, where $\mathbf{x}$ accounts for the first modality, $\mathbf{y}$ for the second one, and $\mathbf{z}$ for the supervision labels. Now, we assume that there exists two functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{y})$ which take $\mathbf{x}$ and $\mathbf{y}$ as inputs, and output $\hat{\mathbf{z}}_\mathbf{x}$ and $\hat{\mathbf{z}}_\mathbf{y}$, which are estimates of the ground-truth labels $\mathbf{z}$.

Furthermore, functions $\mathbf{f}$ and $\mathbf{g}$ are composed of $M$ and $N$ layers, respectively, subfunctions denoted by $\mathbf{f}_l$ and $\mathbf{g}_l$. With a slight abuse of notation, we write for layer $l$, $\mathbf{x}_l = (\mathbf{f}_l \circ \mathbf{f}_{l-1} \cdots \circ \mathbf{f}_1)(\mathbf{x})$, and $\mathbf{y}_l = (\mathbf{g}_l \circ \mathbf{g}_{l-1} \cdots \circ \mathbf{g}_1)(\mathbf{y})$. See Fig. 1 for a visual representation. Examples of subfunctions when dealing with standard neural networks are operations like convolution, pooling, multiplication by a matrix, non-linearity, etc. The outputs of these subfunctions are the features we want to fuse across modalities. The problem is then to choose which features to fuse and how to mix them.

### 3.1. Multimodal fusion search space

In our approach, data fusion is introduced through a third neural network (see Fig. 2 for some illustrations). Each fusion layer $l$ combines three inputs: the output of the previous fusion layer and one output from each modality. This is done according to the following equation:

$$\mathbf{h}_l = \boldsymbol{\sigma}_{\gamma_l^p} \left( \mathbf{W}_l \begin{bmatrix} \mathbf{x}_{\gamma_l^m} \\ \mathbf{y}_{\gamma_l^n} \\ \mathbf{h}_{l-1} \end{bmatrix} \right) \tag{1}$$

where $\boldsymbol{\gamma}_l = (\gamma_l^m, \gamma_l^n, \gamma_l^p)$ is a triplet of variable indices establishing, respectively, which feature from the first modality, which feature from the second modality, and which non-linearity is applied. Also, $\gamma_l^m \in \{1, \cdots, M\}$, $\gamma_l^n \in \{1, \cdots, N\}$, and $\gamma_l^p \in \{1, \cdots, P\}$. For the first fusion layer ($l = 1$), the fusion operation is defined as:

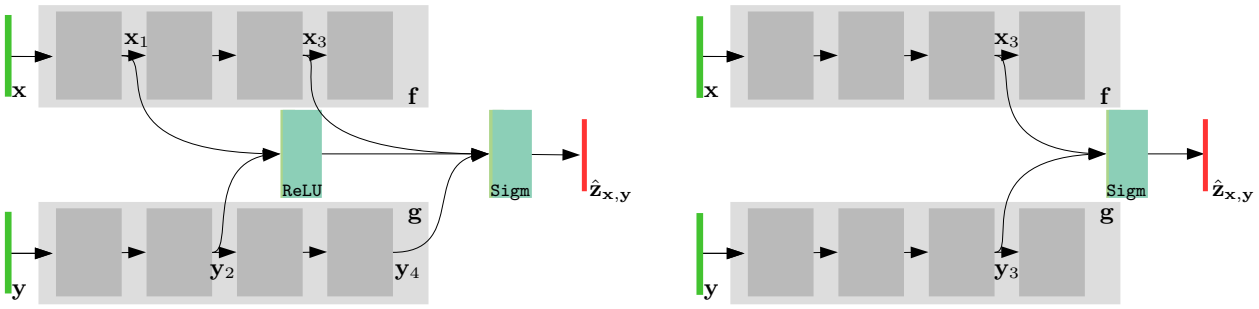


Figure 2. **Two realizations of our search space on a small bimodal network**. Left: network defined by $[(\gamma_1^m = 1, \gamma_1^n = 2, \gamma_1^p = 1), (\gamma_2^m = 3, \gamma_2^n = 4, \gamma_2^p = 2)]$. Right: network defined by $[(\gamma_1^m = 3, \gamma_1^n = 3, \gamma_1^p = 2)]$.

$$\mathbf{h}_1 = \boldsymbol{\sigma}_{\gamma_1^p} \left( \mathbf{W}_1 \begin{bmatrix} \mathbf{x}_{\gamma_1^m} \\ \mathbf{y}_{\gamma_1^n} \end{bmatrix} \right) \tag{2}$$

The number of possible fusion layers, a search parameter, is denoted by $L$, so that $l \in \{1, \cdots, L\}$. The fusion layer weight matrix $\mathbf{W}_l$ is trainable. Note that we establish feature concatenation as fixed strategy to process and fuse features. In fact, this could be replaced by a weighted sum of input features. However, during our experiments, we noticed that fusion networks with weighted sum of features were almost never chosen, and almost always reduced final classification performance with respect to concatenation. Thus, we decided to simply fix the fusion operation to concatenation.

An illustrative example for $M = N = 4$, and $P = 2$ ($p = 1$ : `ReLU`; $p = 2$ : `Sigmoid`) is shown in Fig. 2. We can observe a couple of realizations of the search space for modalities of four hidden layers and two possible non-linearities. On the right, a fusion scheme with a single fusion at the third layer of first and second modalities. On the left, two composed fusions. A composed fusion scheme is defined then by a vector of triplets: $[\boldsymbol{\gamma}_l]_{l \in \{1, \cdots, L\}}$. We denote the set of all possible triplets with $L$ layers as $\boldsymbol{\Gamma}_L$.

Observe that this design enables our space to contain a large number of possible fusion architectures, including the networks defined in, for example, CentralNet [42]. The size of the search space is exponential on the number of fusion layers $L$, and is expressed by: $(M \times N \times P)^L$. If we were to tackle a multimodal problem where the number of layers of the feature extractor is only a portion of the depth that modern neural networks exhibit, say $M = N = 16$, and only considered two possible non-linearities $P = 2$, a fusion scheme with $L = 5$ would result in a search space of dimension $\sim \mathbf{3, 51 \times 10^{13}}$.

Exhaustively exploring all these possibilities is intractable. In particular, consider that evaluation of a single sample in this space corresponds to training and evaluating a multimodal architecture, which can take from several hours to a few days, depending on the problem at hand. This is the reason why we focus on an exploration method that has shown to be sampling-efficient for the related problem of neural architecture search. This is, sequential model-based optimization (SMBO), as used by [26, 34]. In their works, the authors showed that progressively exploring a search space by dividing it into "complexity levels", ends up providing architectures that perform as well as the ones discovered by a more direct exploration approach, as in [48, 49], while sampling fewer architectures. SMBO is well fit to find optimal architectures in the search space designed by [49]. This is because the space is naturally divided by complexity levels that can be interpreted as progression steps (blocks in the "micro space" [26, 34, 49]). SMBO sequentially unfolds the complexity of the sampled architectures starting from the simplest one. Luckily, our search space shares a similar structure. We can interpret the number of fusion layers $L$ as a hook for progression.

It is worth noting that the constrained search space that we propose exhibits certain desirable properties. Assuming that the unimodal feature extractor networks are available greatly reduces search burden as they do not need to be trained during search, and the complexity of the problem is limited to a manageable magnitude.

### 3.2. Search algorithm

In SMBO, a model predicting accuracy of sampled architectures lies at the core of the method. This model, or *surrogate* function is trained during progressive exploration of the search space, and it is used to reduce the amount of neural networks that have to be trained and evaluated by predicting performance of unseen architectures. In our case, having a variable-length description of the multimodal architectures $[\boldsymbol{\gamma}_l]_{l \in \{1, \cdots, L\}}$, as described in previous subsection, naturally results in using a recurrent model as *surrogate*. Let us denote this recurrent function by $\boldsymbol{\pi}$. The parameters of $\boldsymbol{\pi}$ are updated at iteration $l$ by stochastic gradient descent (SGD) training on a subset of $\boldsymbol{\Gamma}_l$ with real valued accuracies $\mathcal{A}_l$.

**Algorithm 1** Multimodal fusion architecture search (MFAS)

1: **procedure** ($\mathbf{f}, \mathbf{g}, \mathrm{L}, \mathrm{E}_{\mathrm{search}}, \mathrm{E}_{\mathrm{train}}, \mathrm{K}, \mathrm{S}_{\mathrm{train}}, \mathrm{S}_{\mathrm{val}}, T_{max}, T_{min}$)
2: L: max number of fusion layers
3: $E_{search}$: number of search iterations
4: $E_{train}$: number of training epochs
5: $K$: number of sampled fusion architectures
6: $S_{train}, S_{val}$: training and validation sets
7: $T_{max}, T_{min}$: sampling temperature range
8: $T \leftarrow \mathrm{T}_{\max}$ *// Set temperature*
9: $\mathcal{B}, \mathcal{A} \leftarrow \{\}$ *// Initialize corresponding sets of arhcs. and accuracy*
10: **for** $e = 1 \cdots \mathrm{E}_{\mathrm{search}}$ **do**
11: $\mathcal{S}_1 \leftarrow \mathbf{\Gamma}_1$ *// Set of fusion architectures with $L = 1$*
12: $\mathcal{M}_1 \leftarrow \texttt{descToFusionNet}(\mathcal{S}_1, \mathbf{f}, \mathbf{g})$ *// Build fusion nets*
13: $\mathcal{C}_1 \leftarrow \texttt{train}(\mathcal{M}_1, \mathrm{S}_{\mathrm{train}}, E_{train})$ *// Train fusion nets*
14: $\mathcal{A}_1 \leftarrow \texttt{evaluate}(\mathcal{C}_1, \mathrm{S}_{\mathrm{val}})$ *// Get real accuracies for them*
15: $\mathcal{B}, \mathcal{A} \leftarrow \mathcal{B} \cup \mathcal{S}_1, \mathcal{A} \cup \mathcal{A}_1$ *// Keep track of sampled archs.*
16: $\boldsymbol{\pi} \leftarrow \texttt{update}(\mathcal{S}_1, \mathcal{A}_1)$ *// Train surrogate*
17: **for** $l = 2 \cdots L$ **do**
18: $\mathcal{S}'_l \leftarrow \texttt{addLayer}(\mathcal{S}_{l-1}, \mathbf{\Gamma}_l)$ *// Unfold 1 more fusion layer*
19: $\hat{\mathcal{A}}'_l \leftarrow \texttt{pred}(\mathcal{S}'_l, \boldsymbol{\pi})$ *// Predict with surrogate*
20: $\mathcal{P}_l \leftarrow \texttt{computeProbs}(\hat{\mathcal{A}}'_l, T)$ *// Compute sampling probs.*
21: $\mathcal{S}_l \leftarrow \texttt{sampleK}(\mathcal{S}'_l, \mathcal{P}_l, K)$ *// Sample K fusion archs*
22: $\mathcal{M}_l \leftarrow \texttt{descToFusionNet}(\mathcal{S}_l, \mathbf{f}, \mathbf{g})$ *// Build fusion net.*
23: $\mathcal{C}_l \leftarrow \texttt{train}(\mathcal{M}_l, \mathrm{S}_{\mathrm{train}}, E_{train})$ *// Train*
24: $\mathcal{A}_l \leftarrow \texttt{evaluate}(\mathcal{C}_l, \mathrm{S}_{\mathrm{val}})$ *// Calculate accuracies*
25: $\mathcal{B}, \mathcal{A} \leftarrow \mathcal{B} \cup \mathcal{S}_l, \mathcal{A} \cup \mathcal{A}_l$ *// Keep track of sampled archs.*
26: $\boldsymbol{\pi} \leftarrow \texttt{update}(\mathcal{S}_l, \mathcal{A}_l)$ *// Update surrogate*
27: $T \leftarrow \texttt{updateTemperature}(T, T_{max}, T_{min})$
28: **end for**
29: **end for**
**return** $\texttt{topK}(\mathcal{B}, \mathcal{A}, K)$ *// Return best K from all sampled archs.*
30: **end procedure**

Our procedure, named *multimodal fusion architecture search* (MFAS), and based on [26], is laid out in Alg. 1. From lines 11 to 16, the progressive algorithm starts at the smallest fusion network complexity level, *i.e.*, $L = 1$. Then, the next complexity levels unroll one after the other by sampling $K$ architectures with a probability that is a function of the surrogate model predictions in lines 20 and 21. The fusion architecture search is effectively guided by how new architectures are sampled. Observe that we implement search iterations ($\mathrm{E}_{\mathrm{search}}$) and temperature-based sampling ($\mathrm{T}_{\max}, \mathrm{T}_{\min}$) as in EPNAS [34]. This is done so the surrogate function does not guide the search with biased assumptions made from partial observations of the search space at early iterations. By using temperature-based sampling, the surrogate function is only trusted as the exploration advances (by reducing the temperature in line 27). This is complemented by training sampled architectures with very few epochs as in ENAS [35], and implementing weight-sharing among sampled architectures to counterweight the main bottleneck of neural architecture search: training sampled architectures to completion. This aspect is of particular importance for multimodal networks, which tend to have a large memory footprint and computing times.

Another aspect where our search algorithm differs from the original algorithm [26] and from [34] is that we assume the existence of pre-trained modal functions $\mathbf{f}$ and $\mathbf{g}$. These functions are used to build a multimodal network from a description of the fusion scheme $\mathcal{S}_l$ with $l$ layers (line 12 and line 22). At the end of the iterative progressive search, MFAS returns the best $K$ from the set of all sampled architectures $\mathcal{B}$.

**Final architecture.** From Alg. 1, we obtain a set of $K$ fusion architectures. One could think of using the surrogate function after its last update to predict the very best fusion scheme from those. However, in this paper we train the best five of the final $K$ architectures to completion, and simply pick the absolute best one from the obtained validation accuracies. During this last training step we also evaluate the performance of the chosen architectures with a larger size of matrices $\mathbf{W}_l$. The reduced size is used during search to improve sampling speed and to reduce memory costs.

**Loss function.** During the search, the weights of the feature extractors $\mathbf{f}$ and $\mathbf{g}$ are frozen. Because of it, only the fusion softmax $\hat{\mathbf{z}}_{x,y}$ is used for the loss function. Found architectures are initially trained for a few epochs with frozen $\mathbf{f}$ and $\mathbf{g}$ functions. A second training step with more epochs involves a multitask loss on $\hat{\mathbf{z}}_x$, $\hat{\mathbf{z}}_y$, $\hat{\mathbf{z}}_{x,y}$, and unfrozen $\mathbf{f}$ and $\mathbf{g}$ functions. A categorical cross-entropy loss is used in all the reported experiments unless otherwise noted.

**Handling arbitrary tensor dimensions.** A practical issue during the creation of a multimodal neural network from $\mathbf{f}$ and $\mathbf{g}$ is that subfunctions might deliver tensors with arbitrary dimensions, hindering fusion of arbitrary modalities and layer positions. To deal with this in a generic way, we perform global pooling along the channel dimension of 2D and 3D convolutions, while leaving linear layer outputs as they are.

As a side note, observe in Eq. 1 that our default layer type for fusion is fully connected. We experimented with several forms of 1D convolutions without noticing any improvements.

**Weight sharing of fusion layers.** In our implementation of Alg. 1, multimodal neural networks are not trained in parallel. Instead, sampled fusion networks are trained sequentially for a small number of epochs ($E_{train} = 2$ in all of our experiments). For two sample indices $s$ and $s'$, where $s' > s$, we keep track of the weight matrix $\mathbf{W}_l^s$ for layer $l$, so $\mathbf{W}_l^{s'}$ is initialized from $\mathbf{W}_l^s$ if $\texttt{sizeof}(\mathbf{W}_l^s) = \texttt{sizeof}(\mathbf{W}_l^{s'})$. Please note that weights are only shared among matrices in the same layer $l$.

Table 1. Evaluation of our search method on the AV-MNIST dataset. The fusion architectures described by arrays of numbers are instances of our search space with $M = 3, N = 5, P = 2$. Validation accuracy is reported.

| Method | Modalities | Acc |
|---|---|---|
| Top-5 found architectures by random search | | |
| $[(3,3,2),(5,3,2)]$ | image + spect. | 0.9174 |
| $[(1,1,2),(4,3,1),(5,2,1)]$ | image + spect. | 0.9190 |
| $[(5,3,1),(4,1,2)]$ | image + spect. | 0.9196 |
| $[(5,2,1),(5,3,1)]$ | image + spect. | 0.9224 |
| $[(5,3,1)]$ | image + spect. | 0.9222 |
| Mean (Std) | 0.9203 (0.0021) | |
| Top-5 found architectures by MFAS | | |
| $[(3,3,2),(5,2,1),(1,3,1),(1,1,2)]$ | image + spect. | 0.9258 |
| $[(5,2,1),(5,2,2),(5,1,1)]$ | image + spect. | 0.9260 |
| $[(5,3,1),(4,2,1),(5,3,1)]$ | image + spect. | **0.9270** |
| $[(5,3,1),(4,2,1),(3,3,2)]$ | image + spect. | 0.9266 |
| $[(4,3,1),(5,3,1),(4,3,1),(5,3,1)]$ | image + spect. | 0.9268 |
| Mean (Std) | 0.9264 (0.0004) | |

## 4. Experiments

In this section we present an extensive experimental validation of our claims. We first start by presenting experiments on a synthetic toy dataset, namely the AV-MNIST dataset [42]. We then continue our experimental work by directly tackling two other multimodal datasets. These are i) the visual-textual multilabel movie genre classification dataset by [3] (MM-IMDB) and ii) the multimodal action recognition dataset by [37] (NTU RGB+D).

For each dataset, we provide a short description of the task as well as the experimental set-up, and then discuss on the results.

AV-MNIST **dataset.** This is a simple audio-visual dataset artificially assembled from independent visual and audio datasets. The first modality corresponds to $28 \times 28$ *MNIST* images, with 75% of their energy removed by PCA. The audio modality is made of audio samples on which we have computed $112 \times 112$ spectrograms. The audio samples are 25,102 pronounced digits of the *Tidigits* database augmented by adding randomly chosen *noise* samples from the *ESC-50* dataset [36]. Contaminated audio samples are randomly paired, accordingly with labels, with MNIST digits in order to reach 55,000 pairs for training and 10,000 pairs for testing. For validation we take 5000 samples from the training set. The digit energy removal and audio contamination are intentionally done to increase the difficulty of the task (otherwise unimodal networks would achieve almost perfect results and data fusion would not be necessary).

In here, $\mathbf{f}$ function is a modified LeNet network [23] with five convolutional layers and a global pooling softmax processing spoken digits. Similarly $\mathbf{g}$ is a modified LeNet with three convolutional layers. We limit the subfunctions of $\mathbf{f}$ and $\mathbf{g}$ to convolutional layers with `ReLU` activation, so we hook global pooling to each one of them: three for the written digit modality ($N = 3$), and five for the spectrogram one ($M = 5$). For this experiment we let $P = 2$ by allowing the activation functions of fusion layers to be either `ReLU` or `Sigmoid`.

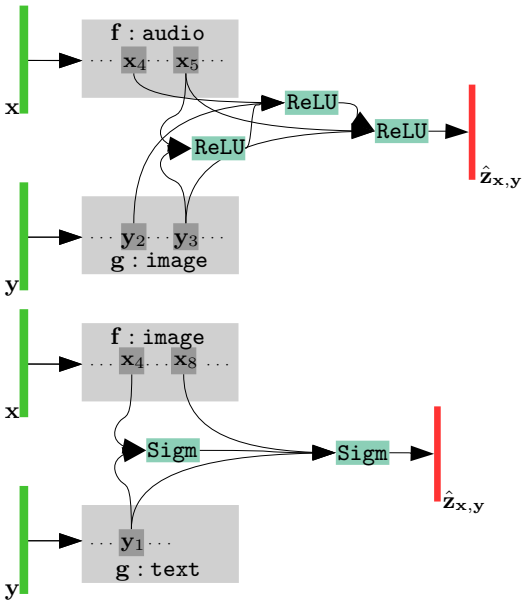


Figure 3. **Structure of the found fusion architectures**. First: AV-MNIST. Second: MM-IMDB.

Table 2. Evaluation of multiple fusion architectures on the AV-MNIST dataset. Test accuracy is reported.

| Method | Modalities | Acc (%) |
|---|---|---|
| Unimodal baselines for fusion | | |
| LeNet-3 [23] | image | 74.52 |
| LeNet-5 [23] | spectrogram | 66.06 |
| Explicit fusion | | |
| Two-stream [39] | image + spect. | 87.78 |
| CentralNet [42] | image + spect. | 87.86 |
| Ours Top 1 | image + spect. | **88.38** |

In Table 1, we show results for two exploration approaches: a purely random one (upper part), and MFAS (bottom). Both exploration approaches are allowed to sample 180 architectures. We show validation accuracy for the top five randomly sampled architectures on the proposed search space (top of Table 1). The large standard deviation is a testament to the usefulness of multimodal fusion architecture search. From these results we can infer that some feature combinations provide better insights from data than some other mixtures. At the lower part of Table 1 we can see that in contrast to random search, the top five found architectures with our search method present scores with less variability. Furthermore, the best performing architecture on the validation set (in bold) is found by our method.

Test accuracy for baselines and competing fusion architectures are reported in Table 2. We report test score of our best found architecture according to Table 1. It can be ob-

Table 3. Evaluation of multiple methods on the MM-IMDB dataset [37]. Weighted F1 (F1-W) and Macro F1 (F1-M) are reported for each method.

| Method | Modalities | F1-W | F1-M |
|---|---|---|---|
| Unimodal baselines for fusion | | | |
| Maxout MLP [12] | text | 0.5754 | 0.4598 |
| VGG Transfer | image | 0.4921 | 0.3350 |
| Explicit fusion | | | |
| Two-stream [39] | image + text | 0.6081 | 0.5049 |
| GMU [3] | image + text | 0.6170 | 0.5410 |
| CentralNet [42] | image + text | 0.6223 | 0.5344 |
| Ours Top 1 | image + text | **0.6250** | **0.5568** |

served that all multimodal fusion networks largely improve over the unimodal networks, but our automatically found fusion architecture is the one with the best overall score. This was found after three iterations of progressive search and $L = 4$. The success on this toy (but not trivial) dataset is a first milestone in the validation of our contributions.

MM-IMDB **dataset.** This multimodal dataset comprises 25,959 movie titles and metadata from the *Internet Movie Database*[1] [3]. Movie data is formed by their plots, posters (RGB images), genres, and many more metadata fields including director, writer, picture format, etc. The task in this dataset is to predict movie genres from posters and movie descriptions. Since very often a movie is assigned to more than one genre, the classification is multi-label. The loss function used for training is binary cross-entropy with weights to balance the dataset.

The original split of the dataset is used in our experiments: 15,552 movies are used for training, 7,799 for testing, and 2,608 for validation. The genres to predict include *drama*, *comedy*, *documentary*, *sport*, *western*, *film-noir*, etc., for a total of 23 non-mutually exclusive classes.

Performance of unimodal networks is given at the top of Table 3. Using these unimodal networks as a basis, we implemented Two-stream fusion [39], CentralNet [42], GMU [3], and our best found architecture. One can note that our method gives the best results among the four fusion strategies, once again validating our choices on search space design and fusion scheme [2].

The search space for the MM-IMDB dataset is formed by eight convolutional layers of a VGG-19 image network, and two text Maxout-MLP features. The number of possible fusion configurations available from these features (we set $N = 2$, and $M = 8$) and the three possible non-linearities (`ReLU`, `Sigmoid`,and `LeakyReLU`) is of 110,592. Our best configuration can be seen in Fig. 3.

[1] `https://www.imdb.com/`

[2] The original Central-Net paper considers only the last feature layer for each modality (as pre-computed by the original authors [3]). Intermediate layers were not available to us. Consequently, we did not start with the exact same unimodal baselines and re-implemented all methods in order to allow fair comparison.

Table 4. Evaluation of multiple methods on the NTU RGB+D dataset [37]. The reported numbers are the average accuracy over the different action subjects (cross-subject measure).

| Method | Modalities | Acc (%) |
|---|---|---|
| Single modality | | |
| LSTM [37] | pose | 60.69 |
| part-LSTM [37] | pose | 62.93 |
| Spatio-temp. attention [41] | pose | 73.40 |
| Multiple modalities | | |
| Shahroudy *et al.* [38] | video + pose | 74.86 |
| Shahroudy *et al.* [38] | video + pose | 74.86 |
| Bilinear Learning [17] | video + pose | 83.30 |
| Bilinear Learning [17] | video + pose + depth | 85.40 |
| 2D/3D Multitask [28] | video + pose | 85.50 |
| Unimodal baselines for fusion | | |
| Inflated ResNet-50 [7] | video | 83.91 |
| Co-occurrence [24] | pose | 85.24 |
| Explicit fusion | | |
| Two-stream [39] | video + pose | 88.60 |
| GMU [3] | video + pose | 85.80 |
| CentralNet [42] | video + pose | 89.36 |
| Ours Top 1 | video + pose | **90.04**$\pm$0.6 |

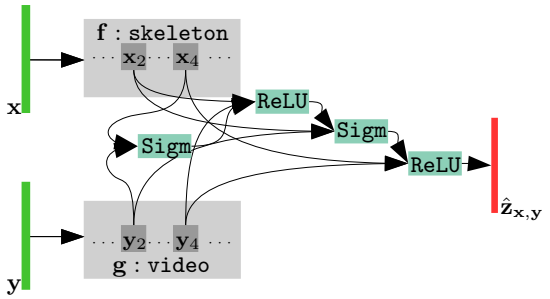


Figure 4. **Structure of found fusion architectures**. NTU RGB+D.

NTU RGB+D **dataset.** This dataset was first introduced by Shahroudy *et al.*, [37] in 2016. With 56,880 samples, to the best of our knowledge, it is the largest color and depth multimodal dataset. Capturing 40 subjects from 80 viewpoints performing 60 classes of activities NTU RGB+D is a very challenging dataset with the particularity that it provides dynamic skeleton-based pose data on the top of RGB video sequences. The target activities include *drinking*, *eating*, *falling down*, and even subject interactions like *hugging*, *shaking hands*, *punching*, etc.

| Network | # of fusion Parameters | Acc (%) |
|---|---|---|
| 0 | 2,229,248 | 0.9327 |
| 1 | 2,196,480 | 0.9289 |
| 2 | 1,737,728 | 0.9301 |
| 3 | 2,163,712 | **0.9346** |

Table 5. Top 4 found architectures on NTU RGB+D according to validation accuracy during search.

In our work, we focus on the cross-subject evaluation, splitting the 40 subjects into training, validation, and testing

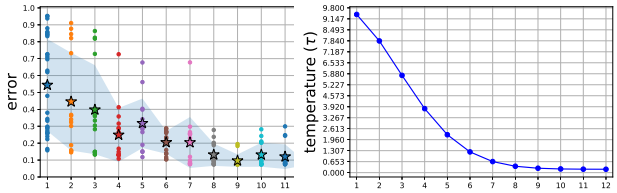


Figure 5. **Left: Error progression during search**. Each plot point represents the validation error of a sampled fusion architecture at a given step of our search algorithm on the AV-MNIST set, where the total number of steps is $\mathrm{E}_{\mathrm{search}} * L$. Mean error and standard deviation per step are represented with stars and plot shadow, respectively. **Right: search temperature schedule**.

groups. The subject IDs for training during search are: 1, 4, 8, 13, 15, 17, 19. For validation we use: 2, 5, 9, and, 14. During final training of the found architectures we use the same splitting originally proposed by [37]. We report results on the testing set to objectively compare our found architectures with manually designed fusion strategies from the state-of-the-art.

Results on the NTU RGB+D dataset are summarized in Table 4. We report accuracy in percentages for several methods. The first group of methods are models processing single modalities as reported by the authors themselves. The second group of results are by methods from the state-of-the-art processing and fusing several modalities (video, pose, and/or depth). Then, we provide the score as computed by us of methods processing single modalities. For video, we tested the *Inflated ResNet-50* used by [7]; and for pose, we leverage the deep co-occurrence model by [24]. The reported numbers in this group are our departing point and baselines. Finally, the last group of methods perform explicit fusion of modalities and are our main competitors.

Observe that our scores are the highest in Table 4. We report $90.04\%$ average accuracy over four runs with a variance of $0.6$, which is a significant improvement over all baselines and competing methods. This is achieved by performing fusion search on the convolutional and fully connected features of the Inflated ResNet-50 and deep Co-occurrence baselines. We start from four possible features for each modality ($M = N = 4$) and three non-linearities, *i.e.*, `ReLU`, `Sigmoid`,and `LeakyReLU`. This means, the search space for the NTU RGB+D dataset is of dimension $5,308,416$. The best found configuration is shown in Fig. 4. In Table 5 we report validation accuracy during search for the final top four architectures. Observe that the best architecture is not necessarily the largest one.

**Multimodal fusion search behaviour.** In Fig. 5 (top) we display the behaviour of our search procedure by plotting validation errors of sampled architectures. It can be observed that, overall, sampled architectures are more and more stable error-wise as the search progresses. The stabilization of sampled errors originates from two sources: first, the shared fusion weights have been more refined at the final steps of the search, and second, the search is driven with more confidence by the predictions of the surrogate function. Indeed, at the last few steps, mean error is significantly lower than the initial ones.

Another interesting effect of our search method and fusion scheme is the fact that even at the initial search steps it is possible to sample architectures that display relatively small validation errors. Since the fusion weights of sampled architectures are trained only for a few epochs, this effect is not necessarily a positive reflection of how good or bad the sampled architecture is. Indeed, it is possible to sample a simple fusion scheme on very deep uni-modal features (which have been pretrained offline) and outperform other sampled architectures that might actually perform better when its weights are revisited at later search steps. In this sense, our temperature-driven sampling of architectures offers a way to escape the fake local minima that originate from this phenomenon. This all boils down to the fact that it is important, in order to avoid getting trapped by initial biased evidence, to trust the surrogate function only after exploration has advanced. We use an inverse exponential schedule for the sampling temperature, as shown at the bottom of Fig. 5, since we observed a better outcome in comparison to a linear temperature schedule.

Table 6. Search timings and hardware configurations.

| Dataset | GPUs (P100) | $\mathrm{E}_{\mathrm{search}} * L$ (steps) | Search time (hours) | Avg. step time (hours) |
|---|---|---|---|---|
| AV-MNIST | 1 | $3 * 4 = 12$ | 3.42 | 0.285 |
| MM-IMDB | 1 | $5 * 3 = 15$ | 9.24 | 0.616 |
| NTU RGB+D | 4 | $3 * 4 = 12$ | 150.91 | 12.57 |

**Search timings.** In Table 6 we provide the hardware settings and timings for the search on all the reported datasets. Multi GPU training through data parallelism was necessary on the NTU RGB+D. Search times on NTU RGB+D are much larger than on the MM-IMDB dataset due to model complexity and larger search space.

## 5. Conclusion

This work tackles the problem of finding accurate fusion architectures for multimodal classification. We propose a novel multimodal search space and exploration algorithm to solve the task in an efficient yet effective manner. The proposed search space is constrained in such a way that it allows convoluted architectures to take place while also containing the complexity of the problem to reasonable levels. We experimentally demonstrated on three datasets the validity of our method, discovering several fusion schemes that provide state-of-the-art results on those datasets. Future research directions include improving the search space so the composition of fusion layers is even more flexible.

## References


[1] M. R. Amer, T. Shields, B. Siddiquie, A. Tamrakar, A. Divakaran, and S. Chai. Deep multimodal fusion: A hybrid approach. In *IJCV*. Springer, 2018.

[2] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *ICML*, 2013.

[3] J. Arevalo, T. Solorio, M. Montes-y Gómez, and F. A. González. Gated multimodal units for information fusion. In *ICLR Workshop*, 2017.

[4] P. K. Atrey, M. A. Hossain, A. El Saddik, and M. S. Kankanhalli. Multimodal fusion for multimedia analysis: a survey. *Multimedia Systems*, 16(6):345–379, 2010.

[5] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential deep learning for human action recognition. In *ECCV Workshop*, pages 29–39. Springer, 2011.

[6] F. R. Bach, G. R. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *ICML*. ACM, 2004.

[7] F. Baradel, C. Wolf, J. Mille, and G. W. Taylor. Glimpse clouds: Human activity recognition from unstructured feature points. In *CVPR*, volume 3, 2018.

[8] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome. Mutan: Multimodal tucker fusion for visual question answering. In *ICCV*, volume 3, 2017.

[9] A. Brock, T. Lim, J. M. Ritchie, and N. Weston. Smash: one-shot model architecture search through hypernetworks. In *ICLR*, 2017.

[10] M. Engilberge, L. Chevallier, P. Pérez, and M. Cord. Finding beans in burgers: Deep semantic-visual embedding with localization. In *CVPR*, 2018.

[11] V. Escorcia, J. Carlos Niebles, and B. Ghanem. On the relationship between visual attributes and convolutional networks. In *CVPR*, pages 1256–1264, 2015.

[12] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, and Y. Bengio. Maxout networks. In *ICML*, 2013.

[13] Z. Gu, B. Lang, T. Yue, and L. Huang. Learning joint multimodal representation based on multi-fusion deep neural networks. In *ICONIP*, 2017.

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[15] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine*, 29(6):82–97, 2012.

[16] C. Hori, T. Hori, T.-Y. Lee, Z. Zhang, B. Harsham, J. R. Hershey, T. K. Marks, and K. Sumi. Attention-based multimodal fusion for video description. In *ICCV*, 2017.

[17] J.-F. Hu, W.-S. Zheng, J. Pan, J. Lai, and J. Zhang. Deep bilinear learning for rgb-d action recognition. In *ECCV*, pages 335–351, 2018.

[18] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

[19] F. Hutter, H. H. Hoos, and K. Leyton-Brown. Sequential model-based optimization for general algorithm configuration. In *International Conference on Learning and Intelligent Optimization*, pages 507–523. Springer, 2011.

[20] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 1991.

[21] Y.-G. Jiang, Z. Wu, J. Wang, X. Xue, and S.-F. Chang. Exploiting feature and class relationships in video categorization with regularized deep neural networks. *T-PAMI*, 2018.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[23] Y. LeCun, B. E. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. E. Hubbard, and L. D. Jackel. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1990.

[24] C. Li, Q. Zhong, D. Xie, and S. Pu. Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation. *arXiv preprint arXiv:1804.06055*, 2018.

[25] F. Li, N. Neverova, C. Wolf, and G. Taylor. Modout: Learning to fuse modalities via stochastic regularization. In *CVIS*, 2016.

[26] C. Liu, B. Zoph, J. Shlens, W. Hua, L.-J. Li, L. Fei-Fei, A. Yuille, J. Huang, and K. Murphy. Progressive neural architecture search. In *ECCV*, 2018.

[27] X. Long, C. Gan, G. de Melo, X. Liu, Y. Li, F. Li, and S. Wen. Multimodal keyless attention fusion for video classification. In *AAAI*, 2018.

[28] D. C. Luvizon, D. Picard, and H. Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *CVPR*, volume 2, 2018.

[29] P. Natarajan, S. Wu, S. Vitaladevuni, X. Zhuang, S. Tsakalidis, U. Park, R. Prasad, and P. Natarajan. Multimodal feature fusion for robust event detection in web videos. In *CVPR*, 2012.

[30] N. Neverova, C. Wolf, G. Taylor, and F. Nebout. Moddrop: adaptive multi-modal gesture recognition. *T-PAMI*, 2016.

[31] N. Neverova, C. Wolf, G. W. Taylor, and F. Nebout. Multi-scale deep learning for gesture detection and localization. In *ECCV Workshop*, pages 474–490. Springer, 2014.

[32] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, pages 483–499. Springer, 2016.

[33] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, 2011.

[34] J.-M. Pérez-Rúa, M. Baccouche, and S. Pateux. Efficient progressive neural architecture search. In *BMVC*, 2018.

[35] H. Pham, M. Y. Guan, B. Zoph, Q. V. Le, and J. Dean. Efficient neural architecture search via parameter sharing. In *ICML*, 2018.

[36] K. J. Piczak. Esc: Dataset for environmental sound classification. In *International Conference on Multimedia*, 2015.

[37] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *CVPR*, pages 1010–1019, 2016.

[38] A. Shahroudy, T.-T. Ng, Y. Gong, and G. Wang. Deep multimodal feature analysis for action recognition in rgb+ d videos. *T-PAMI*, 2017.

[39] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, pages 568–576, 2014.

[40] C. G. Snoek, M. Worring, and A. W. Smeulders. Early versus late fusion in semantic video analysis. In *ACMM*, 2005.

[41] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, volume 1, pages 4263–4270, 2017.

[42] V. Vielzeuf, A. Lechervy, S. Pateux, and F. Jurie. Centralnet: a multilayer approach for multimodal fusion. In *ECCV Workshop*, 2018.

[43] Y. Wu, E. Y. Chang, K. C.-C. Chang, and J. R. Smith. Optimal multimodal fusion for multimedia data analysis. In *ACMM*, 2004.

[44] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*, 2016.

[45] X. Yang, P. Molchanov, and J. Kautz. Multilayer and multimodal fusion of deep neural networks for video classification. In *ACMM*, pages 978–987, 2016.

[46] G. Ye, D. Liu, I.-H. Jhuo, and S.-F. Chang. Robust late fusion with rank minimization. In *CVPR*, 2012.

[47] X. Zhou and B. Bhanu. Feature fusion of side face and gait for video-based human identification. *Pattern Recognition*, 2008.

[48] B. Zoph and Q. V. Le. Neural architecture search with reinforcement learning. In *ICLR*, 2017.

[49] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *CVPR*, 2018.