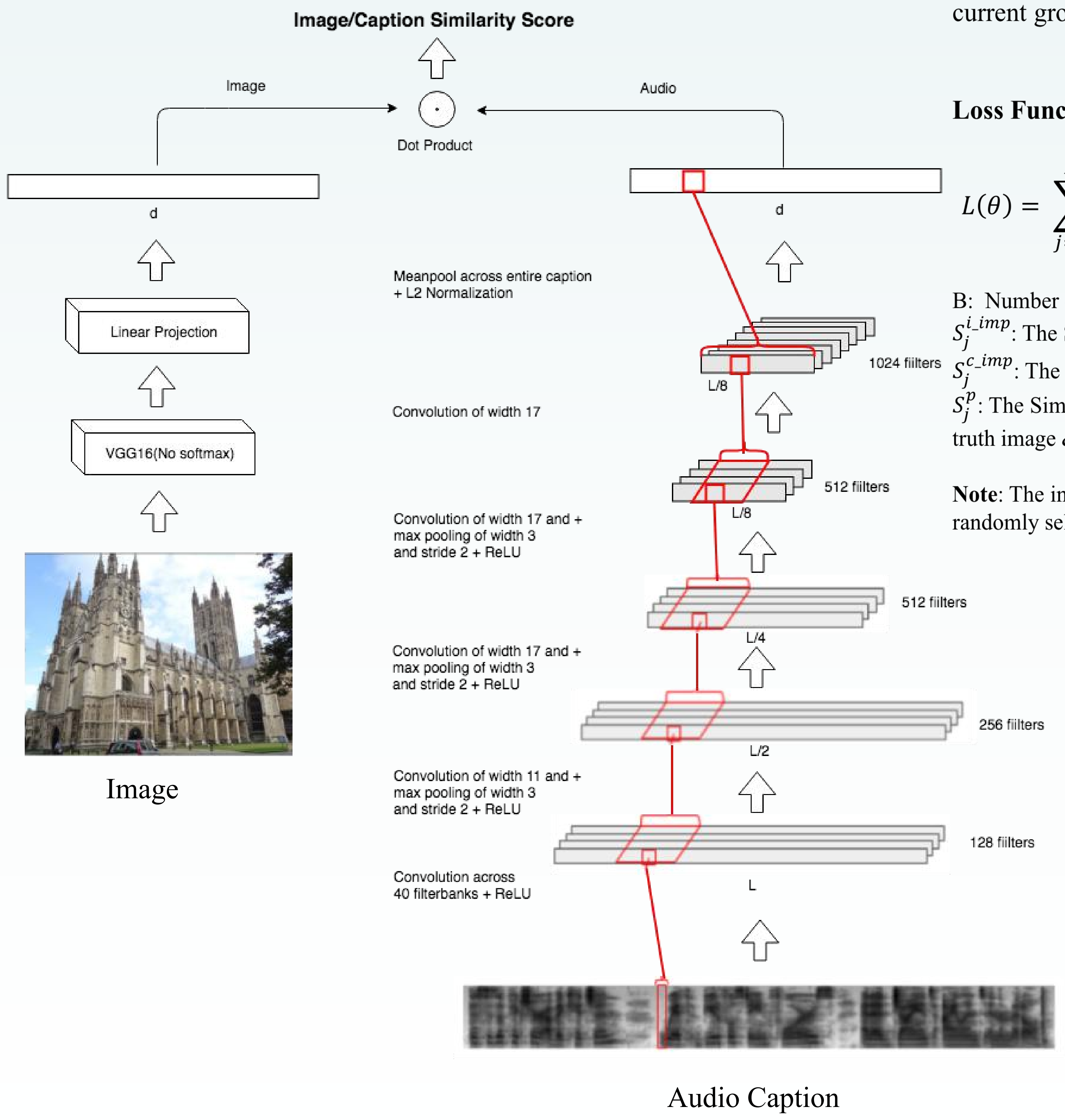
Optimization Of Multimodal Analysis Of Image And Audio Pairs
CS598PS-Signal Processing For Machine Learning

Junrui Ni, Yuxi Gu, Xuanyi Zhu
University of Illinois at Urbana-Campaign

Abstract

Given a dataset of images and spoken audio captions, the paper written by Harwath et al. (2016) presents a method for discovering word-like acoustic units from continuous speech at the waveform level with no additional text transcriptions or conventional speech recognition apparatus and grounding them to semantically relevant image regions. Based on their work, we extend and optimize their model to achieve better performance and accuracy to find a better match between image and audios. And our optimized machine learning algorithm can be used to search images based on audio captions or search things in the opposite way.

Model Architecture



Baseline Model

Image Branch:
Taking VGG 16 layer network(weights fixed) and replacing the softmax classification layer with a linear transform which maps the 4096-dimensional activations of the second fully connected layer into our 1024-dimensional multimodal embedding space.

Audio Caption Branch:
Transform the audio segments into speech spectrograms and feed into a customized convolutional neural network as shown below.

Connection Part:
The overall multimodal network is formed by tying together the learned visual feature vectors and learned audio feature vectors and map them into a shared embedding space. Then, computes an inner product between them, representing the similarity score between a given image/caption pair.

Terms:
Imposter Image/Caption: An image/caption that is randomly selected from the current batch, but is its content is different from the current ground truth image/caption.

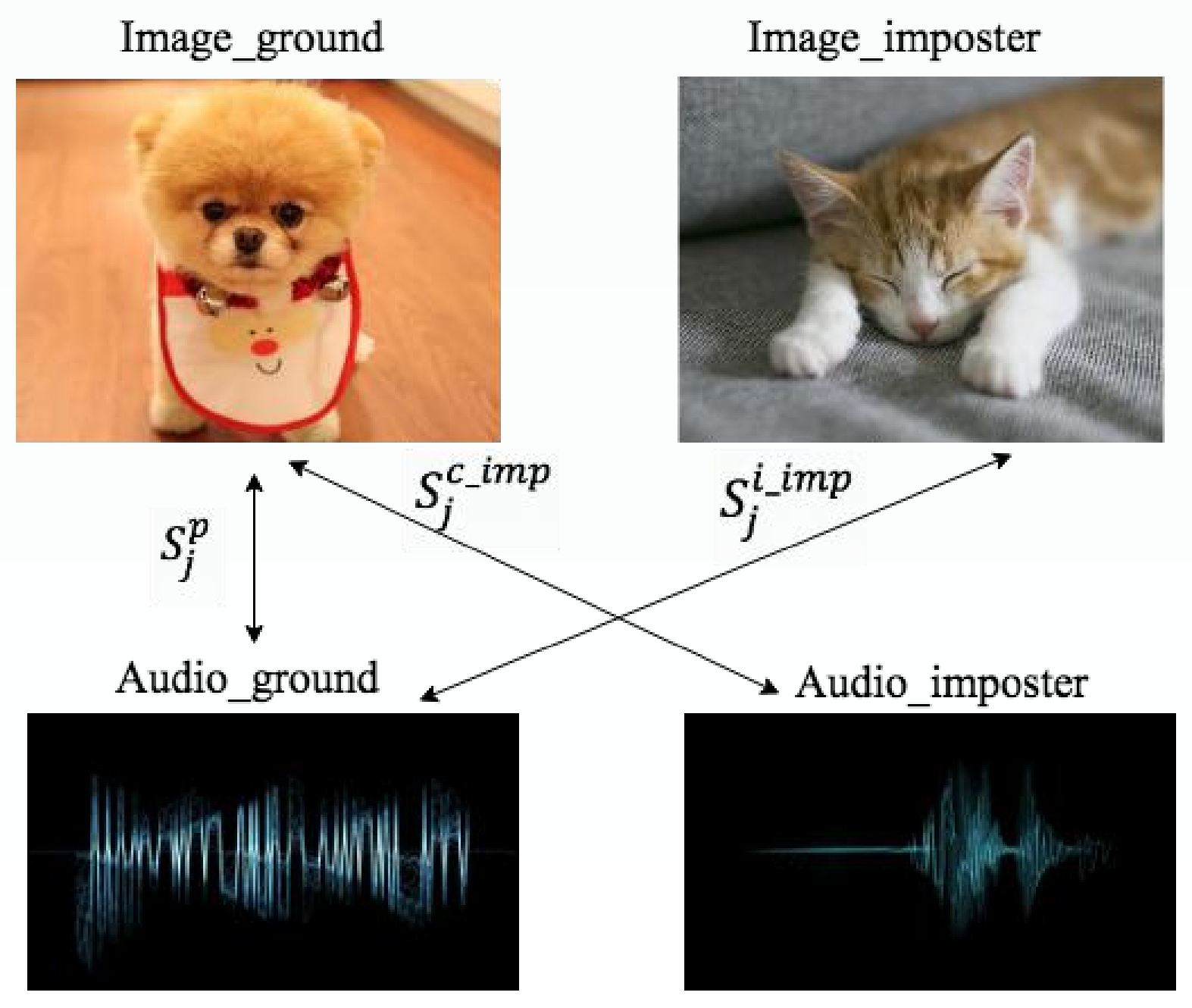
Loss Function:

$$L(\theta) = \sum_{j=1}^{B} \max\left(0, S_j^{c\_imp} - S_j^{p} + 1\right) + \max(0, S_j^{i\_imp} - s_j^{p} + 1)$$

B: Number of Image/Caption pairs.
$S_j^{i\_imp}$: The Similarity score between j^{th} caption and its imposter image.
$S_j^{c\_imp}$: The Similarity score between j^{th} image and its imposter caption.
S_j^{p}: The Similarity score between j^{th} image and j^{th} caption, which is the ground truth image & audio caption pair.

Note: The index of the imposter image and the index of the imposter caption are randomly selected and they can be different.

Visualization of Loss Function



Optimization

Modification of Loss Function
The basic idea is inspired by the original loss function. But we will find the matched image for a selected imposter caption and find the matched caption for a selected image for calculating loss function. And we modified the loss function to be following:

$$L(\theta) = \sum_{j=1}^{B} t1 + t2 + t3$$

B: Number of Image/Caption pairs.

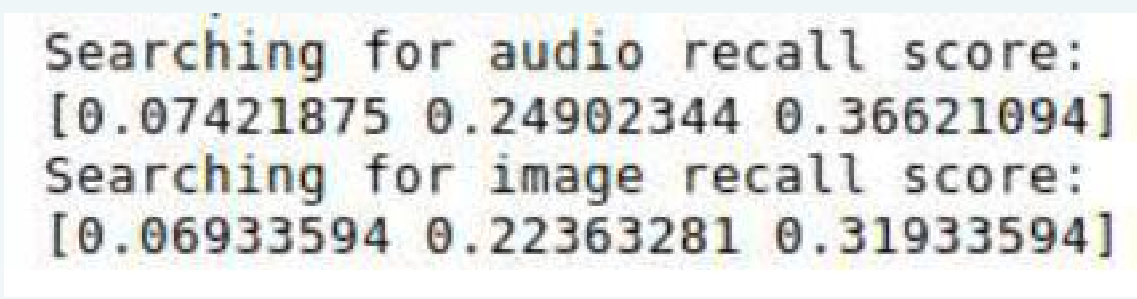
$$t1 = \max\left(0, S_j^{imp1Caption\_gImage} - S_j^{gCaption\_gImage} + 1\right) + \max(0,\ S_j^{gCatption\_imp2Image} - s_j^{gCaption\_gImage} + 1)$$

$$t2 = \max\left(0, S_j^{gCaption\_imp1Image} - S_j^{imp1Caption\_imp1Image} + 1\right) + \max(0, S_j^{imp1Caption\_gImage} - s_j^{imp1Caption\_imp1Image} + 1)$$

$$t3 = \max\left(0, S_j^{gCaption\_imp2Image} - S_j^{imp2Caption\_imp2Image} + 1\right) + \max(0, S_j^{imp2Caption\_gImage} - s_j^{imp2Caption\_imp2Image} + 1)$$

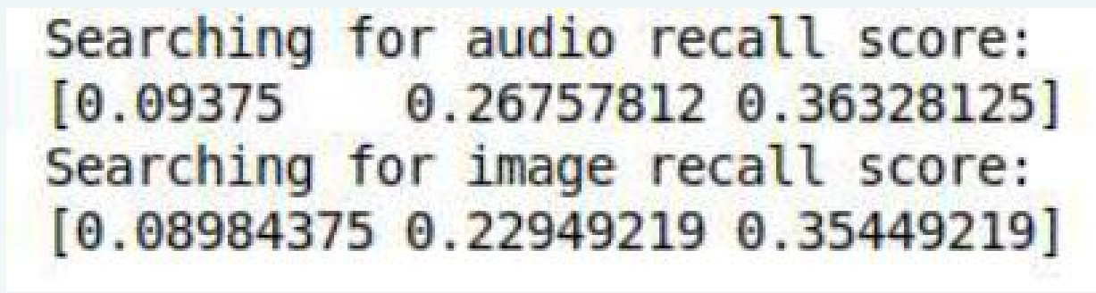
$S_j^{ACaption\_BImage}$: The Similarity score between A's caption and B's image when we are using j^{th} image/caption pair as the ground truth image/caption pair.

Comparison:



```
Searching for audio recall score:
[0.07421875 0.24902344 0.36621094]
Searching for image recall score:
[0.06933594 0.22363281 0.31933594]
```

Baseline



```
Searching for audio recall score:
[0.09375    0.26757812 0.36328125]
Searching for image recall score:
[0.08984375 0.22949219 0.35449219]
```

Result Using New Loss Function

Modification of Model Architecture
We proposed to augment the spectrogram feature vector by explicitly modelling temporal information in another branch parallel with the baseline architecture. Suppose our spectrogram is an image of size (num_freq, num_time, num_channel). First, three sets of vertical frequency filters are learned to collapse the frequency axis. Then another set of filters are learned for local temporal information. Horizontal max pooling is then applied, before the output is fed into the next layer, which is called "Temporal Convolution Networks".

This kind of network is basically a 1D fully convolutional network with three modifications, causal convolution, dilation and residual block. Causal convolution is used to model the relationship f(yt|yt-1...yt-k), while dilation is used to exponentially increase the receptive field. Residual block is used between previous layer and current layer to learn modifications to the identity mapping, rather than full transformation.

An illustration of a single TCN stack is shown below.

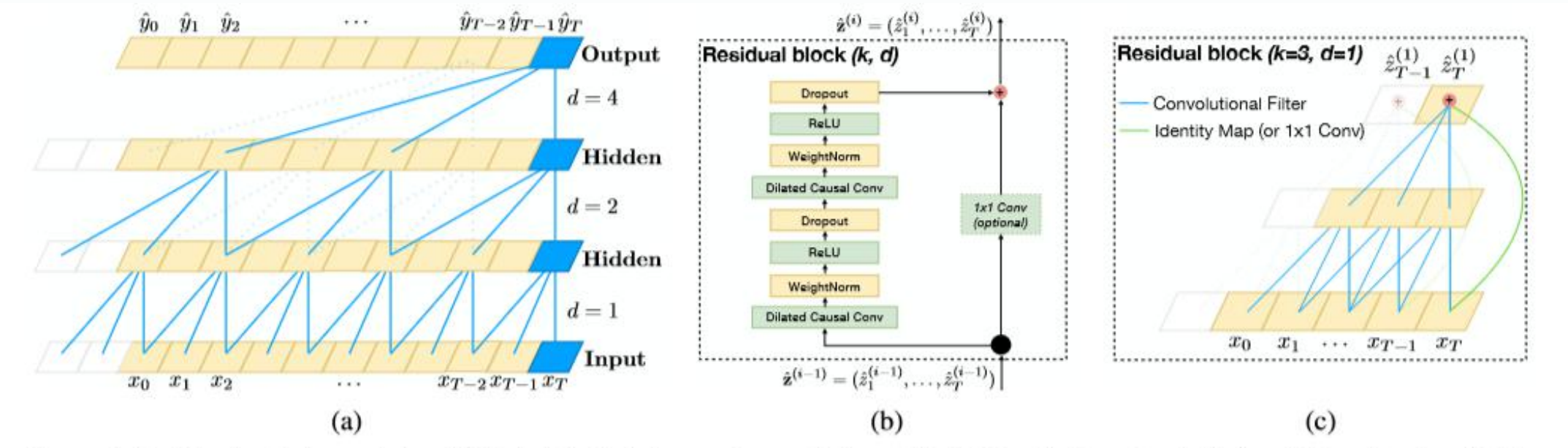

$\hat{y}_0$ $\hat{y}_1$ $\hat{y}_2$ ... $\hat{y}_{T-2}$ $\hat{y}_{T-1}$ $\hat{y}_T$
Output
$d = 4$
Hidden
$d = 2$
Hidden
$d = 1$
Input
x_0 x_1 x_2 ... x_{T-2} x_{T-1} x_T
Residual block (k, d)
Dropout
ReLU
WeightNorm
Dilated Causal Conv
Dropout
ReLU
WeightNorm
Dilated Causal Conv
1x1 Conv (optional)
$\hat{z}^{(i)} = (\hat{z}_1^{(i)}, \dots, \hat{z}_T^{(i)})$
$\hat{z}^{(i-1)} = (\hat{z}_1^{(i-1)}, \dots, \hat{z}_T^{(i-1)})$
Residual block (k=3, d=1)
$\hat{z}_{T-1}^{(1)}$ $\hat{z}_T^{(1)}$
Convolutional Filter
Identity Map (or 1x1 Conv)
x_0 x_1 ... x_{T-1} x_T
(a)
(b)
(c)

Figure 1. Architectural elements in a TCN. (a) A dilated causal convolution with dilation factors $d = 1, 2, 4$ and filter size $k = 3$. The receptive field is able to cover all values from the input sequence. (b) TCN residual block. An 1x1 convolution is added when residual input and output have different dimensions. (c) An example of residual connection in a TCN. The blue lines are filters in the residual function, and the green lines are identity mappings.

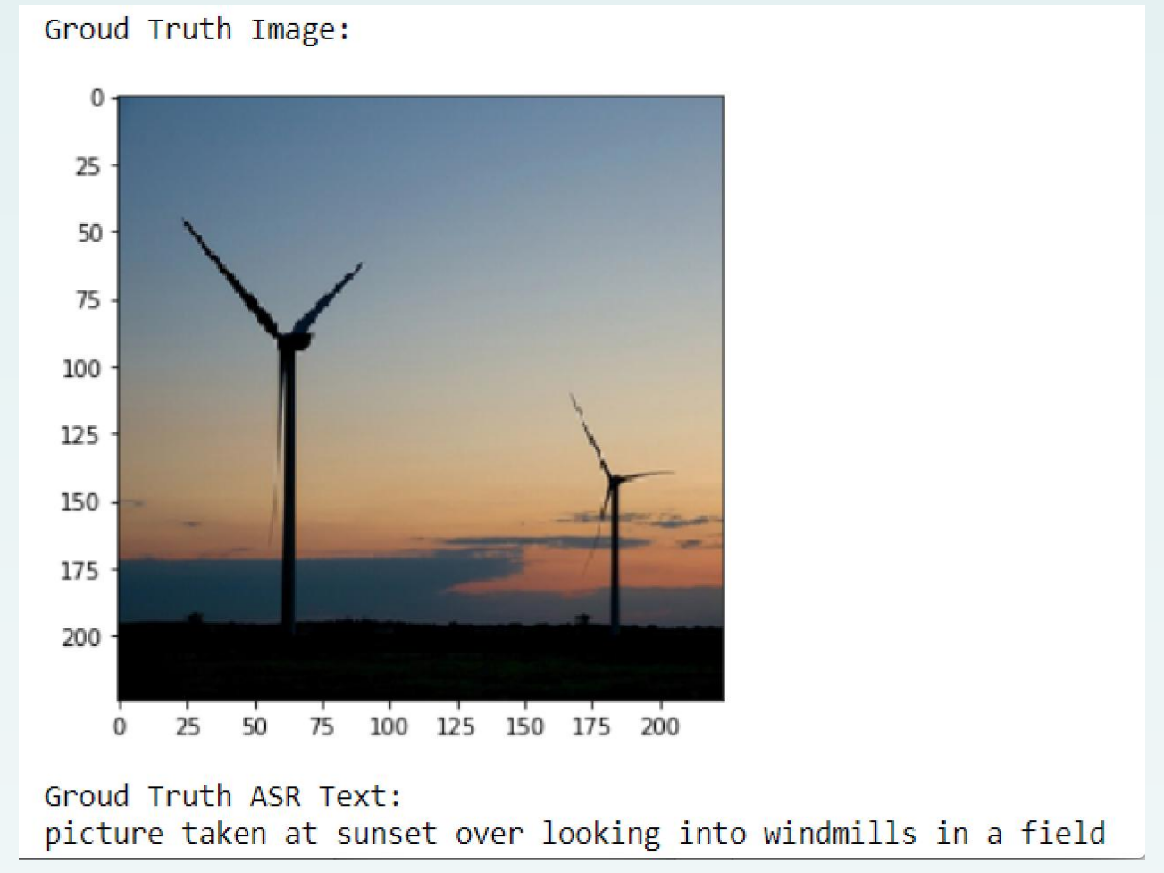
We used two stacks of TCN, which, combined, gives a total receptive field of over the whole temporal dimension, so the last temporal vector of the second TCN stack is used as our temporal feature vector. This temporal vector is concatenated with the caption-wised pooled vector of the baseline architecture, with the number of feature maps reduced. The concatenated feature vector is then fed through a dense layer, in the hope to learn a meaningful correlation among the concatenated vector.

Results

There are two ways to use our developed machine learning model.
1) The input query is an image, and the output is related captions.
2) The input query is a caption, and the output is related images.

For better understanding and visualization, we transform audio captions into ASR text captions and also provide ground truth images with returned text captions in pairs here. And an example of search result is below. Shown on the top is an image with its ground truth audio transformed into text caption form. Below are its five highest scoring audio files transformed into text caption form.

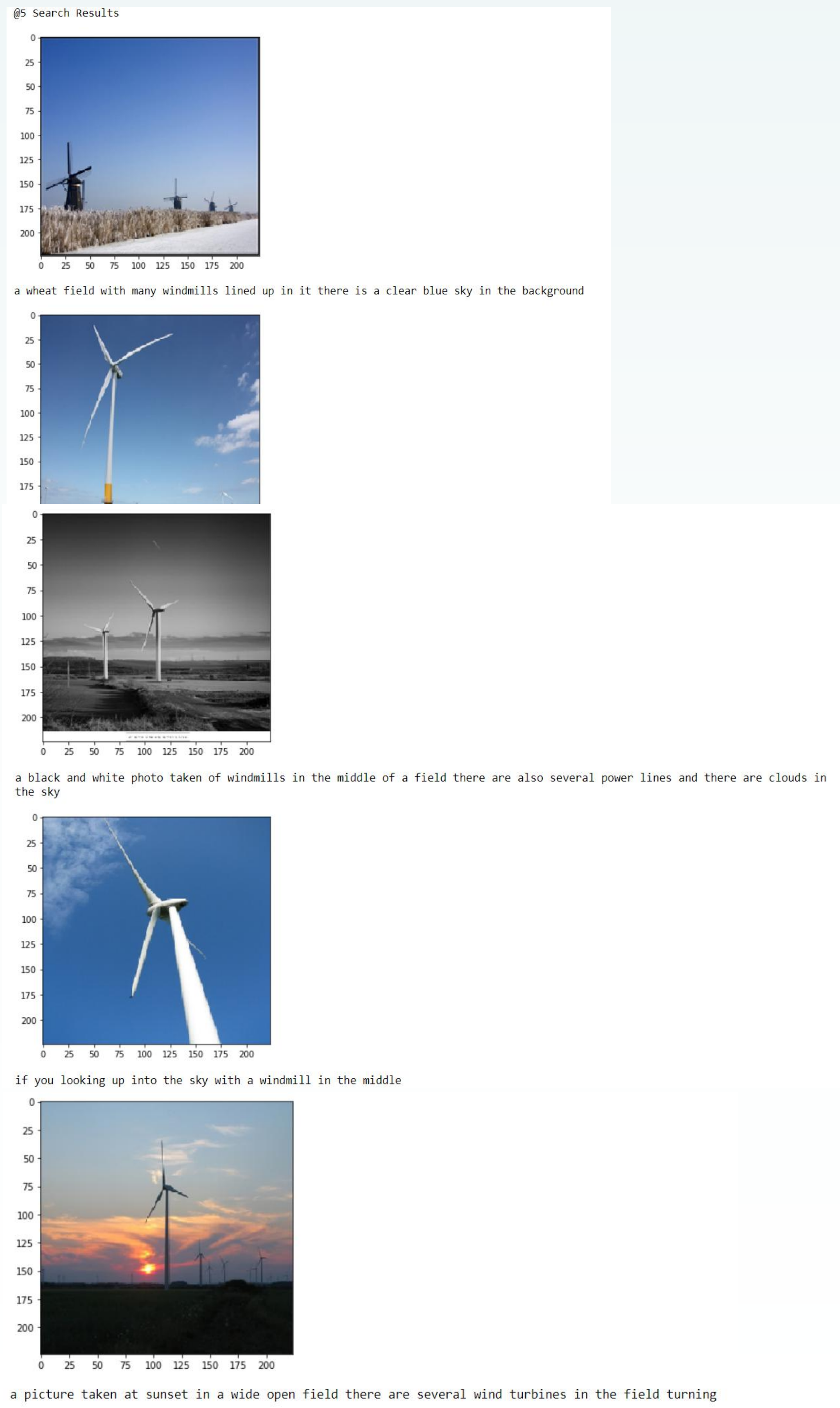
Input Image:



Groud Truth Image:

Groud Truth ASR Text:
picture taken at sunset over looking into windmills in a field

Search Results:



@5 Search Results

a wheat field with many windmills lined up in it there is a clear blue sky in the background

a black and white photo taken of windmills in the middle of a field there are also several power lines and there are clouds in the sky

if you looking up into the sky with a windmill in the middle

a picture taken at sunset in a wide open field there are several wind turbines in the field turning