Identifying Visible Actions in Lifestyle Vlogs

Oana Ignat[1], Laura Burdick[1], Jia Deng[2], Rada Mihalcea[1]
[1]University of Michigan, [2]Princeton University
{oignat,wenlaura,mihalcea}@umich.edu, jiadeng@cs.princeton.edu

Abstract

We consider the task of identifying human actions visible in online videos. We focus on the widely spread genre of lifestyle vlogs, which consist of videos of people performing actions while verbally describing them. Our goal is to identify if actions mentioned in the speech description of a video are visually present. We construct a dataset with crowdsourced manual annotations of visible actions, and introduce a multimodal algorithm that leverages information derived from visual and linguistic clues to automatically infer which actions are visible in a video. We demonstrate that our multimodal algorithm outperforms algorithms based only on one modality at a time.

1 Introduction

There has been a surge of recent interest in detecting human actions in videos. Work in this space has mainly focused on learning actions from articulated human pose (Du et al., 2015; Vemulapalli et al., 2014; Zhang et al., 2017) or mining spatial and temporal information from videos (Simonyan and Zisserman, 2014; Wang et al., 2016). A number of resources have been produced, including Action Bank (Sadanand and Corso, 2012), NTU RGB+D (Shahroudy et al., 2016), SBU Kinect Interaction (Yun et al., 2012), and PKU-MMD (Liu et al., 2017).

Most research on video action detection has gathered video information for a set of pre-defined actions (Fabian Caba Heilbron and Niebles, 2015; Real et al., 2017; Kay et al., 2017), an approach known as *explicit data gathering* (Fouhey et al., 2018). For instance, given an action such as "open door," a system would identify videos that include a visual depiction of this action. While this approach is able to detect a specific set of actions, whose choice may be guided by downstream applications, it achieves high precision at the cost of

low recall. In many cases, the set of predefined actions is small (e.g., 203 activity classes in Fabian Caba Heilbron and Niebles 2015), and for some actions, the number of visual depictions is very small.

An alternative approach is to start with a set of videos, and identify all the actions present in these videos (Damen et al., 2018; Bregler, 1997). This approach has been referred to as *implicit data gathering,* and it typically leads to the identification of a larger number of actions, possibly with a small number of examples per action.

In this paper, we use an implicit data gathering approach to label human activities in videos. To the best of our knowledge, we are the first to explore video action recognition using both transcribed audio and video information. We focus on the popular genre of lifestyle vlogs, which consist of videos of people demonstrating routine actions while verbally describing them. We use these videos to develop methods to identify if actions are visually present.

The paper makes three main contributions. First, we introduce a novel dataset consisting of 1,268 short video clips paired with sets of actions mentioned in the video transcripts, as well as manual annotations of whether the actions are visible or not. The dataset includes a total of 14,769 actions, 4,340 of which are visible. Second, we propose a set of strong baselines to determine whether an action is visible or not. Third, we introduce a multimodal neural architecture that combines information drawn from visual and linguistic clues, and show that it improves over models that rely on one modality at a time.

By making progress towards automatic action recognition, in addition to contributing to video understanding, this work has a number of important and exciting applications, including sports analytics (Fani et al., 2017), human-computer inter-

Dataset	#Actions	#Verbs	#Actors	Implicit	Label types
Ours	4340	580	10	✓	✓
VLOG (Fouhey et al., 2018)	-	-	10.7k	✓	✓
Kinetics (Kay et al., 2017)	600	270	-	x	x
ActivityNet (Fabian Caba Heilbron and Niebles, 2015)	203	-	-	x	x
MIT (Monfort et al., 2019)	339	339	-	x	x
AVA (Gu et al., 2018)	80	80	192	✓	x
Charades (Sigurdsson et al., 2016)	157	30	267	x	x
MPII Cooking (Rohrbach et al., 2012)	78	78	12	✓	x

Table 1: Comparison between our dataset and other video human action recognition datasets. # Actions show either the number of action classes in that dataset (for the other datasets), or the number of unique visible actions in that dataset (ours); # Verbs shows the number of unique verbs in the actions; Implicit is the type of data gathering method (versus explicit); Label types are either post-defined (first gathering data and then annotating actions): ✓, or pre-defined (annotating actions before gathering data): x.

action (Rautaray and Agrawal, 2015), and automatic analysis of surveillance video footage (Ji et al., 2012).

The paper is organized as follows. We begin by discussing related work, then describe our data collection and annotation process. We next overview our experimental set-up and introduce a multimodal method for identifying visible actions in videos. Finally, we discuss our results and conclude with general directions for future work.

2 Related Work

There has been substantial work on action recognition in the computer vision community, focusing on creating datasets (Soomro et al., 2012; Karpathy et al., 2014; Sigurdsson et al., 2016; Fabian Caba Heilbron and Niebles, 2015) or introducing new methods (Herath et al., 2017; Carreira and Zisserman, 2017; Donahue et al., 2015; Tran et al., 2015). Table 1 compares our dataset with previous action recognition datasets.[1]

The largest datasets that have been compiled to date are based on YouTube videos (Fabian Caba Heilbron and Niebles, 2015; Real et al., 2017; Kay et al., 2017). These actions cover a broad range of classes including human-object interactions such as cooking (Rohrbach et al., 2014; Das et al., 2013; Rohrbach et al., 2012) and playing tennis (Karpathy et al., 2014), as well as human-human interactions such as shaking hands and hugging (Gu et al., 2018).

Similar to our work, some of these previous datasets have considered everyday routine actions (Fabian Caba Heilbron and Niebles, 2015; Real et al., 2017; Kay et al., 2017). However, because these datasets rely on videos uploaded on YouTube, it has been observed they can be potentially biased towards unusual situations (Kay et al., 2017). For example, searching for videos with the query "drinking tea" results mainly in unusual videos such as dogs or birds drinking tea. This bias can be addressed by paying people to act out everyday scenarios (Sigurdsson et al., 2016), but this can end up being very expensive. In our work, we address this bias by changing the approach used to search for videos. Instead of searching for actions in an explicit way, using queries such as "opening a fridge" or "making the bed," we search for more general videos using queries such as "my morning routine." This approach has been referred to as implicit (as opposed to explicit) data gathering, and was shown to result in a greater number of videos with more realistic action depictions (Fouhey et al., 2018).

Although we use implicit data gathering as proposed in the past, unlike (Fouhey et al., 2018) and other human action recognition datasets, we search for routine videos that contain rich audio descriptions of the actions being performed, and we use this transcribed audio to extract actions. In these lifestyle vlogs, a vlogger typically performs an action while also describing it in detail. To the best of our knowledge, we are the first to build a video action recognition dataset using both transcribed audio and video information.

Another important difference between our

---

[1]Note that the number of actions shown for our dataset reflects the number of unique visible actions in the dataset and not the number of action classes, as in other datasets. This is due to our annotation process (see §3).

methodology and previously proposed methods is that we extract action labels from the transcripts. By gathering data before annotating the actions, our action labels are post-defined (as in Fouhey et al. 2018). This is unlike the majority of the existing human action datasets that use pre-defined labels (Sigurdsson et al., 2016; Fabian Caba Heilbron and Niebles, 2015; Real et al., 2017; Kay et al., 2017; Gu et al., 2018; Das et al., 2013; Rohrbach et al., 2012; Monfort et al., 2019). Post-defined labels allow us to use a larger set of labels, expanding on the simplified label set used in earlier datasets. These action labels are more inline with everyday scenarios, where people often use different names for the same action. For example, when interacting with a robot, a user could refer to an action in a variety of ways; our dataset includes the actions "stick it into the freezer," "freeze it," "pop into the freezer," and "put into the freezer," variations, which would not be included in current human action recognition datasets.

In addition to human action recognition, our work relates to other multimodal tasks such as visual question answering (Jang et al., 2017; Wu et al., 2017), video summarization (Gygli et al., 2014; Song et al., 2015), and mapping text descriptions to video content (Karpathy and Fei-Fei, 2015; Rohrbach et al., 2016). Specifically, we use an architecture similar to (Jang et al., 2017), where an LSTM (Hochreiter and Schmidhuber, 1997) is used together with frame-level visual features such as Inception (Szegedy et al., 2016), and sequence-level features such as C3D (Tran et al., 2015). However, unlike (Jang et al., 2017) who encode the textual information (question-answers pairs) using an LSTM, we chose instead to encode our textual information (action descriptions and their contexts) using a large-scale language model ELMo (Peters et al., 2018).

Similar to previous research on multimodal methods (Lei et al., 2018; Xu et al., 2015; Wu et al., 2013; Jang et al., 2017), we also perform feature ablation to determine the role played by each modality in solving the task. Consistent with earlier work, we observe that the textual modality leads to the highest performance across individual modalities, and that the multimodal model combining textual and visual clues has the best overall performance.

Query	Results
my morning routine	28M+
my after school routine	13M+
my workout routine	23M+
my cleaning routine	13M+
DIY	78M+

Table 2: Approximate number of videos found when searching for routine and do-it-yourself queries on YouTube.

3 Data Collection and Annotation

We collect a dataset of routine and do-it-yourself (DIY) videos from YouTube, consisting of people performing daily activities, such as making breakfast or cleaning the house. These videos also typically include a detailed verbal description of the actions being depicted. We choose to focus on these lifestyle vlogs because they are very popular, with tens of millions having been uploaded on YouTube; Table 2 shows the approximate number of videos available for several routine queries. Vlogs also capture a wide range of everyday activities; on average, we find thirty different visible human actions in five minutes of video.

By collecting routine videos, instead of searching explicitly for actions, we do *implicit* data gathering, a form of data collection introduced by Fouhey et al. 2018. Because everyday actions are common and not unusual, searching for them directly does not return many results. In contrast, by collecting routine videos, we find many everyday activities present in these videos.

3.1 Data Gathering

We build a data gathering pipeline (see Figure 1) to automatically extract and filter videos and their transcripts from YouTube. The input to the pipeline is manually selected YouTube channels. Ten channels are chosen for their rich routine videos, where the actor(s) describe their actions in great detail. From each channel, we manually select two different playlists, and from each playlist, we randomly download ten videos.

The following data processing steps are applied:

Transcript Filtering. Transcripts are automatically generated by YouTube. We filter out videos that do not contain any transcripts or that contain transcripts with an average (over the entire video) of less than 0.5 words per second. These videos do not contain detailed action descriptions so we cannot effectively leverage textual information.

Extract Candidate Actions from Transcript.
Starting with the transcript, we generate a noisy list of potential actions. This is done using the Stanford parser (Chen and Manning, 2014) to split the transcript into sentences and identify verb phrases, augmented by a set of hand-crafted rules to eliminate some parsing errors. The resulting actions are noisy, containing phrases such as "found it helpful if you" and "created before up the top you."

Segment Videos into Miniclips. The length of our collected videos varies from two minutes to twenty minutes. To ease the annotation process, we split each video into miniclips (short video sequences of maximum one minute). Miniclips are split to minimize the chance that the same action is shown across multiple miniclips. This is done automatically, based on the transcript timestamp of each action. Because YouTube transcripts have timing information, we are able to line up each action with its corresponding frames in the video. We sometimes notice a gap of several seconds between the time an action occurs in the transcript and the time it is shown in the video. To address this misalignment, we first map the actions to the miniclips using the time information from the transcript. We then expand the miniclip by 15 seconds before the first action and 15 seconds after the last action. This increases the chance that all actions will be captured in the miniclip.

Motion Filtering. We remove miniclips that do not contain much movement. We sample one out of every one hundred frames of the miniclip, and compute the 2D correlation coefficient between these sampled frames. If the median of the obtained values is greater than a certain threshold (we choose 0.8), we filter out the miniclip. Videos with low movement tend to show people sitting in front of the camera, describing their routine, but not acting out what they are saying. There can be many actions in the transcript, but if they are not depicted in the video, we cannot leverage the video information.

3.2 Visual Action Annotation

Our goal is to identify which of the actions extracted from the transcripts are visually depicted in the videos. We create an annotation task on Amazon Mechanical Turk (AMT) to identify actions that are visible.

We give each AMT turker a HIT consisting of five miniclips with up to seven actions generated

Figure 1: Overview of the data gathering pipeline.

from each miniclip. The turker is asked to assign a label (*visible* in the video; *not visible* in the video; *not an action*) to each action. Because it is difficult to reliably separate *not visible* and *not an action*, we group these labels together.

Each miniclip is annotated by three different turkers. For the final annotation, we use the label assigned by the majority of turkers, i.e., *visible* or *not visible / not an action*.

To help detect spam, we identify and reject the turkers that assign the same label for every action in all five miniclips that they annotate. Additionally, each HIT contains a ground truth miniclip that has been pre-labeled by two reliable annotators. Each ground truth miniclip has more than four actions with labels that were agreed upon by both reliable annotators. We compute accuracy between a turker's answers and the ground truth annotations; if this accuracy is less than 20%, we reject the HIT as spam.

After spam removal, we compute the agreement score between the turkers using Fleiss kappa (Fleiss and Cohen, 1973). Over the entire data set, the Fleiss agreement score is 0.35, indicating fair agreement. On the ground truth data, the Fleiss kappa score is 0.46, indicating moderate agreement. This fair to moderate agreement indicates that the task is difficult, and there are cases where the visibility of the actions is hard to label. To illustrate, Figure 3 shows examples where the annotators had low agreement.

Table 3 shows statistics for our final dataset of

Action	Visible?
actually cook it	✓
bake it for	✓
take it out	✓
pull it right off the baking sheet	✓
put it on to some parchment paper	✓
so keep in mind that	x
seems like an eternity in the oven	x
dehydrated at that point which	x

Transcript (left):
...
03:24 you're gonna actually cook it
03:27 and it you're gonna bake it for
03:30 about six hours it's definitely a
03:32 long time so keep in mind that it's
03:34 basically just dehydrating it
03:50 after what seems like an eternity in
03:53 the oven you're going to take it out
03:55 it's actually dehydrated at that point
03:57 which is fabulous because you can
03:59 pull it right off the baking sheet and
04:01 you're going to put it on to some
04:03 parchment paper and then you're
...

Figure 2: Sample video frames, transcript, and annotations.

Videos	177
Video hours	21
Transcript words	302,316
Miniclips	1,268
Actions	14,769
Visible actions	4,340
Non-visible actions	10,429

Table 3: Data statistics.

	Train	Test	Validation
# Actions	11,403	1,999	1,367
# Miniclips	997	158	113
# Actions/ Miniclip	11.4	12.6	12.0

Table 4: Statistics for the experimental data split.

Action	#1	#2	#3	GT
make sure your skin cleansed before you	x	x	✓	x
do all that	✓	x	✓	✓
absorbing all that serum when there	x	x	✓	x
	x	x	✓	x
move on	x	x	x	x

Figure 3: An example of low agreement. The table shows actions and annotations from workers #1, #2, and #3, as well as the ground truth (GT). Labels are: visible - ✓, not visible - x. The bottom row shows screenshots from the video. The Fleiss kappa agreement score is -0.2.

videos labeled with actions, and Figure 2 shows a sample video and transcript, with annotations.

For our experiments, we use the first eight YouTube channels from our dataset as train data, the ninth channel as validation data and the last channel as test data. Statistics for this split are shown in Table 4.

3.3 Discussion

The goal of our dataset is to capture naturally-occurring, routine actions. Because the same action can be identified in different ways (e.g., "pop into the freezer", "stick into the freezer"), our dataset has a complex and diverse set of action labels. These labels demonstrate the language used by humans in everyday scenarios; because of that, we choose not to group our labels into a pre-defined set of actions. Table 1 shows the number of unique verbs, which can be considered a lower bound for the number of unique actions in our dataset. On average, a single verb is used in seven action labels, demonstrating the richness of our dataset.

The action labels extracted from the transcript are highly dependent on the performance of the constituency parser. This can introduce noise or ill-defined action labels. Some acions contain extra words (e.g., "brush my teeth of course"), or lack words (e.g., "let me just"). Some of this noise is handled during the annotation process; for example, most actions that lack words are labeled as "not visible" or "not an action" because they are hard to interpret.

4 Identifying Visible Actions in Videos

Our goal is to determine if actions mentioned in the transcript of a video are visually represented in the video. We develop a multimodal model that leverages both visual and textual information, and we compare its performance with several single-modality baselines.

4.1 Data Processing and Representations

Starting with our annotated dataset, which includes miniclips paired with transcripts and candidate actions drawn from the transcript, we extract several layers of information, which we then use to develop our multimodal model, as well as several baselines.

Action Embeddings. To encode each action, we use both GloVe (Pennington et al., 2014) and ELMo (Peters et al., 2018) embeddings. When using GloVe embeddings, we represent the action as the average of all its individual word embeddings. We use embeddings with dimension 50. When using ELMo, we represent the action as a list of words which we feed into the default ELMo embedding layer.[2] This performs a fixed mean pooling of all the contextualized word representations in each action.

Part-of-speech (POS). We use POS information for each action. Similar to word embeddings (Pennington et al., 2014), we train POS embeddings. We run the Stanford POS Tagger (Toutanova et al., 2003) on the transcripts and assign a POS to each word in an action. To obtain the POS embeddings, we train GloVe on the Google N-gram corpus[3] using POS information from the five-grams. Finally, for each action, we average together the POS embeddings for all the words in the action to form a POS embedding vector.

Context Embeddings. Context can be helpful to determine if an action is visible or not. We use two types of context information, action-level and sentence-level. Action-level context takes into account the previous action and the next action; we denote it as Context_A. These are each calculated by taking the average of the action's GloVe embeddings. Sentence-level context considers up to five words directly before the action and up to five words after the action (we do not consider words that are not in the same sentence as the action);

Action	Con.	Visible?
cook things in **water**	5.00	✓
head right into my **kitchen**	4.97	✓
throw it into the **washer**	4.70	✓
told you what	2.31	x
share my thoughts	2.96	x
prefer them	1.62	x

Table 5: Visible actions with high concreteness scores (Con.), and non-visible actions with low concreteness scores. The noun or verb with the highest concreteness score is in bold.

Action	Visible in the miniclip?
put my son	x
sleep after we	x
done dinner	x
get comfortable	✓
pick out some pajamas	✓
start with my skincare	x
cleanse if I or even	x

we denote it as Context_S. Again, we average the GLoVe embeddings of the preceding and following words to get two context vectors.

Concreteness. Our hypothesis is that the concreteness of the words in an action is related to its visibility in a video. We use a dataset of words with associated concreteness scores from (Brysbaert et al., 2014). Each word is labeled by a human annotator with a value between 1 (very abstract) and 5 (very concrete). The percentage of actions from our dataset that have at least one word in the concreteness dataset is 99.8%. For each action, we use the concreteness scores of the verbs and nouns in the action. We consider the concreteness score of an action to be the highest concreteness score of its corresponding verbs and nouns. Table 5 shows several sample actions along with their concreteness scores and their visiblity.

Video Representations. We use YOLO9000 (Redmon and Farhadi, 2017) to identify objects present in each miniclip. We choose YOLO9000 for its high and diverse number of labels (9,000 unique labels). We sample the miniclips at a rate of 1 frame-per-second, and we use the YOLO9000 model pre-trained on COCO (Lin et al., 2014) and ImageNet (Deng et al., 2009).

We represent a video both at the frame level and the sequence level. For frame-level video features, we use the Inception V3 model (Szegedy

---

[2]Implemented as the ELMo module in Tensorflow
[3]http://storage.googleapis.com/books/ngrams/books/datasetsv2.html

et al., 2016) pre-trained on ImageNet. We extract the output of the very last layer before the Flatten operation (the "bottleneck layer"); we choose this layer because the following fully connected layers are too specialized for the original task they were trained for. We extract Inception V3 features from miniclips sampled at 1 frame-per-second.

For sequence-level video features, we use the C3D model (Tran et al., 2015) pre-trained on the Sports-1M dataset (Karpathy et al., 2014). Similarly, we take the feature map of the sixth fully connected layer. Because C3D captures motion information, it is important that it is applied on consecutive frames. We take each frame used to extract the Inception features and extract C3D features from the 16 consecutive frames around it.

We use this approach because combining Inception V3 and C3D features has been shown to work well in other video-based models (Jang et al., 2017; Carreira and Zisserman, 2017; Kay et al., 2017).

4.2 Baselines

Using the different data representations described in Section 4.1, we implement several baselines.

Concreteness. We label as visible all the actions that have a concreteness score above a certain threshold, and label as non-visible the remaining ones. We fine tune the threshold on our validation set; for fine tuning, we consider threshold values between 3 and 5. Table 6 shows the results obtained for this baseline.

Feature-based Classifier. For our second set of baselines, we run a classifier on subsets of all of our features. We use an SVM (Cortes and Vapnik, 1995), and perform five-fold cross-validation across the train and validation sets, fine tuning the hyper-parameters (kernel type, C, gamma) using a grid search. We run experiments with various combinations of features: action GloVe embeddings; POS embeddings; embeddings of sentence-level context ($Context_S$) and action-level context ($Context_A$); concreteness score. The combinations that perform best during cross-validation on the combined train and validation sets are shown in Table 6.

LSTM and ELMo. We also consider an LSTM model (Hochreiter and Schmidhuber, 1997) that takes as input the tokenized action sequences padded to the length of the longest action. These are passed through a trainable embedding layer,

Figure 4: Example of frames, corresponding actions, object detected with YOLO, and the object - word pair with the highest WUP similarity score in each frame.

initialized with GloVe embeddings, before the LSTM. The LSTM output is then passed through a feed forward network of fully connected layers, each followed by a dropout layer (Srivastava et al., 2014) at a rate of 50%. We use a sigmoid activation function after the last hidden layer to get an output probability distribution. We fine tune the model on the validation set for the number of training epochs, batch size, size of LSTM, and number of fully-connected layers.

We build a similar model that embeds actions using ELMo (composed of 2 bi-LSTMs). We pass these embeddings through the same feed forward network and sigmoid activation function. The results for both the LSTM and ELMo models are shown in Table 6.

YOLO **Object Detection.** Our final baseline leverages video information from the YOLO9000 object detector. This baseline builds on the intuition that many visible actions involve visible objects. We thus label an action as visible if it contains at least one noun similar to objects detected in its corresponding miniclip. To measure similarity, we compute both the Wu-Palmer (WUP) path-length-based semantic similarity (Wu and Palmer, 1994) and the cosine similarity on the GloVe word embeddings. For every action in a miniclip, each noun is compared to all detected objects and assigned a similarity score. As in our concreteness baseline, the action is assigned the highest score of its corresponding nouns. We use the validation data to fine tune the similarity threshold that decides if an action is visible or not. The results are reported in Table 6. Examples of actions that contain one or more words similar to detected objects by YOLO can be seen in Figure 4.
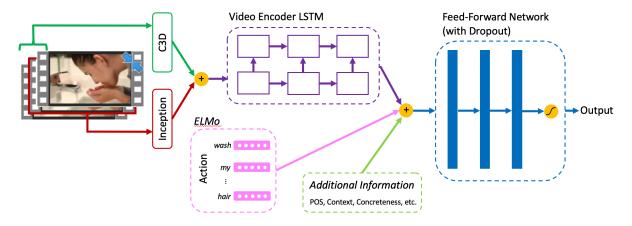
Figure 5: Overview of the multimodal neural architecture. + represents concatenation.

5 Multimodal Model

Each of our baselines considers only a single modality, either text or video. While each of these modalities contributes important information, neither of them provides a full picture. The visual modality is inherently necessary, because it shows the visibility of an action. For example, the same spoken action can be labeled as either *visible* or *non-visible*, depending on its visual context; we find 162 unique actions that are labeled as both visible and not visible, depending on the miniclip. This ambiguity has to be captured using video information. However, the textual modality provides important clues that are often missing in the video. The words of the person talking fill in details that many times cannot be inferred from the video. For our full model, we combine both textual and visual information to leverage both modalities.

We propose a multimodal neural architecture that combines encoders for the video and text modalities, as well as additional information (e.g., concreteness). Figure 5 shows our model architecture. The model takes as input a (miniclip m, action a) pair and outputs the probability that action a is visible in miniclip m. We use C3D and Inception V3 video features extracted for each frame, as described in Section 4.1. These features are concatenated and run through an LSTM.

To represent the actions, we use ELMo embeddings (see Section 4.1). These features are concatenated with the output from the video encoding LSTM, and run through a three-layer feed forward network with dropout. Finally, the result of the last layer is passed through a sigmoid function, which produces a probability distribution indicating whether the action is visible in the mini-

clip. We use an RMSprop optimizer (Tieleman and Hinton, 2012) and fine tune the number of epochs, batch size and size of the LSTM and fully-connected layers.

6 Evaluation and Results

Table 6 shows the results obtained using the multimodal model for different sets of input features. The model that uses all the input features available leads to the best results, improving significantly over the text-only and video-only methods.[4]

We find that using only YOLO to find visible objects does not provide sufficient information to solve this task. This is due to both the low number of objects that YOLO is able to detect, and the fact that not all actions involve objects. For example, visible actions from our datasets such as "get up", "cut them in half", "getting ready", and "chopped up" cannot be correctly labeled using only object detection. Consequently, we need to use additional video information such as Inception and C3D information.

In general, we find that the text information plays an important role. ELMo embeddings lead to better results than LSTM embeddings, with a relative error rate reduction of 6.8%. This is not surprising given that ELMo uses two bidirectional LSTMs and has improved the state-of-the-art in many NLP tasks (Peters et al., 2018). Consequently, we use ELMo in our multimodal model.

Moreover, the addition of extra information improves the results for both modalities. Specifically, the addition of context is found to bring improve-

---

[4]Significance is measured using a paired t-test: $p < 0.005$ when compared to the best text-only model; $p < 0.0005$ when compared to the best video-only model.

Method	Input	Accuracy	Precision	Recall	F1
	BASELINES				
Majority	Action	0.692	0.692	1.0	0.81
Threshold	Concreteness	0.685	0.7	0.954	0.807
Feature-based Classifier	$Action_G$	0.715	0.722	0.956	0.823
	$Action_G$, POS	0.701	0.702	**0.986**	0.820
	$Action_G$, $Context_S$	0.725	0.736	0.938	0.825
	$Action_G$, $Context_A$	0.712	0.722	0.949	0.820
	$Action_G$, Concreteness	0.718	0.729	0.942	0.822
	$Action_G$, $Context_S$, Concreteness	**0.728**	**0.742**	0.932	**0.826**
LSTM	$Action_G$	0.706	0.753	0.857	0.802
ELMo	$Action_G$	**0.726**	**0.771**	**0.859**	**0.813**
YOLO	Miniclip	0.625	0.619	0.448	0.520
	MULTIMODAL NEURAL ARCHITECTURE (FIGURE 5)				
Multi-modal Model	$Action_E$, Inception	0.722	0.765	0.863	0.811
	$Action_E$, Inception, C3D	0.725	0.769	0.869	0.814
	$Action_E$, POS, Inception, C3D	0.731	0.763	0.885	0.820
	$Action_E$, $Context_S$, Inception, C3D	0.725	**0.770**	0.859	0.812
	$Action_E$, $Context_A$, Inception, C3D	0.729	0.757	0.895	0.820
	$Action_E$, Concreteness, Inception, C3D	0.723	0.768	0.860	0.811
	$Action_E$, POS, $Context_S$, Concreteness, Inception, C3D	**0.737**	0.758	**0.911**	**0.827**

Table 6: Results from baselines and our best multimodal method on validation and test data. $Action_G$ indicates action representation using GloVe embedding, and $Action_E$ indicates action representation using ELMo embedding. $Context_S$ indicates sentence-level context, and $Context_A$ indicates action-level context.

ments. The use of POS is also found to be generally helpful.

7 Conclusion

In this paper, we address the task of identifying human actions visible in online videos. We focus on the genre of lifestyle vlogs, and construct a new dataset consisting of 1,268 miniclips and 14,769 actions out of which 4,340 have been labeled as visible. We describe and evaluate several text-based and video-based baselines, and introduce a multimodal neural model that leverages visual and linguistic information as well as additional information available in the input data. We show that the multimodal model outperforms the use of one modality at a time.

A distinctive aspect of this work is that we label actions in videos based on the language that accompanies the video. This has the potential to create a large repository of visual depictions of actions, with minimal human intervention, covering a wide spectrum of actions that typically occur in everyday life.

In future work, we plan to explore additional representations and architectures to improve the accuracy of our model, and to identify finer-grained alignments between visual actions and their verbal descriptions. The dataset and the code introduced in this paper are publicly available at http://lit.eecs.umich.edu/downloads.html.

References

Christoph Bregler. 1997. Learning and recognizing human dynamics in video sequences. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 568–574. IEEE.

Marc Brysbaert, Amy Beth Warriner, and Victor Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior research methods*, 46(3):904–911.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308.

Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. 2018. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736.

Pradipto Das, Chenliang Xu, Richard F Doell, and Jason J Corso. 2013. A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2634–2641.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee.

Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. 2015. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634.

Yong Du, Wei Wang, and Liang Wang. 2015. Hierarchical recurrent neural network for skeleton based action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1110–1118.

Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970.

Mehrnaz Fani, Helmut Neher, David A Clausi, Alexander Wong, and John Zelek. 2017. Hockey action recognition via integrated stacked hourglass network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 29–37.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. 2018. From lifestyle vlogs to everyday interactions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4991–5000.

Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. 2018. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6047–6056.

Michael Gygli, Helmut Grabner, Hayko Riemenschneider, and Luc Van Gool. 2014. Creating summaries from user videos. In *European Conference on Computer Vision (ECCV)*, pages 505–520. Springer.

Samitha Herath, Mehrtash Harandi, and Fatih Porikli. 2017. Going deeper into action recognition: A survey. *Image and vision computing*, 60:4–21.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2758–2766.

Shuiwang Ji, Wei Xu, Ming Yang, and Kai Yu. 2012. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137.

Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. 2014. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. Tvqa: Localized, compositional video question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1369–1379.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pages 740–755. Springer.

Chunhui Liu, Yueyu Hu, Yanghao Li, Sijie Song, and Jiaying Liu. 2017. Pku-mmd: A large scale benchmark for continuous multi-modal human action understanding. *arXiv preprint arXiv:1703.07475*.

Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al. 2019. Moments in time dataset: one million videos for event understanding. *IEEE transactions on pattern analysis and machine intelligence*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Siddharth S Rautaray and Anupam Agrawal. 2015. Vision based hand gesture recognition for human computer interaction: a survey. *Artificial intelligence review*, 43(1):1–54.

Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. 2017. Youtube-boundingboxes: A large high-precision human-annotated data set for object detection in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5296–5305.

Joseph Redmon and Ali Farhadi. 2017. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7263–7271.

Anna Rohrbach, Marcus Rohrbach, Ronghang Hu, Trevor Darrell, and Bernt Schiele. 2016. Grounding of textual phrases in images by reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 817–834. Springer.

Anna Rohrbach, Marcus Rohrbach, Wei Qiu, Annemarie Friedrich, Manfred Pinkal, and Bernt Schiele. 2014. Coherent multi-sentence video description with variable level of detail. In *German conference on pattern recognition*, pages 184–195. Springer.

Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. 2012. A database for fine grained activity detection of cooking activities. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1194–1201. IEEE.

Sreemanananth Sadanand and Jason J Corso. 2012. Action bank: A high-level representation of activity in video. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1234–1241. IEEE.

Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. 2016. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019.

Gunnar A Sigurdsson, Gül Varol, Xiaolong Wang, Ali Farhadi, Ivan Laptev, and Abhinav Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, pages 510–526. Springer.

Karen Simonyan and Andrew Zisserman. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576.

Yale Song, Jordi Vallmitjana, Amanda Stent, and Alejandro Jaimes. 2015. Tvsum: Summarizing web videos using titles. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5179–5187.

Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. 2012. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2):26–31.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*, pages 173–180. Association for Computational Linguistics.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. 2015. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497.

Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. 2014. Human action recognition by representing 3d skeletons as points in a lie group. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 588–595.

Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pages 20–36. Springer.

Qi Wu, Damien Teney, Peng Wang, Chunhua Shen, Anthony Dick, and Anton van den Hengel. 2017. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40.

Qiuxia Wu, Zhiyong Wang, Feiqi Deng, Zheru Chi, and David Dagan Feng. 2013. Realistic human action recognition with multimodal feature selection and fusion. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(4):875–885.

Zhibiao Wu and Martha Palmer. 1994. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics.

Xun Xu, Timothy Hospedales, and Shaogang Gong. 2015. Semantic embedding space for zero-shot action recognition. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 63–67. IEEE.

Kiwon Yun, Jean Honorio, Debaleena Chattopadhyay, Tamara L Berg, and Dimitris Samaras. 2012. Two-person interaction detection using body-pose features and multiple instance learning. In *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 28–35. IEEE.

Pengfei Zhang, Cuiling Lan, Junliang Xing, Wenjun Zeng, Jianru Xue, and Nanning Zheng. 2017. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2117–2126.