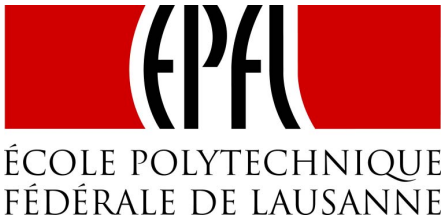
ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE
SCHOOL OF LIFE SCIENCES



Master project in Bioengineering

Human Vision vs. Computer Vision to Classify Simple Actions

Carried out in the Kreiman Laboratory
at Harvard Medical School
Under the supervision of Gabriel Kreiman, Associate Professor

Done by

Vincent JACQUOT

Under the direction of
Professor Michael Herzog
In the Laboratory of Psychophysics (LPSY)

EPFL

Boston and Lausanne, January 19, 2019

Summary

Despite superhuman performance to classify images of any sort of objects, Deep Learning still faces some challenges. One such challenge is to recognize two people interacting, or a person interacting with an object. The current work focused on three simple tasks that were believed difficult to identify for Deep Learning: recognize people reading vs. holding a book, sitting vs. seemingly sitting or drinking vs. holding a glass. For each task, images have to be classified in the category "yes" when the subject in the picture is performing the action, or "no" otherwise.

Three image datasets were created with the preoccupation to obtain high human vision accuracy and low computer vision accuracy. Images in each dataset were ordered, such that images easily classified by convolutional neural networks were discarded. Images were also labelled by several people to obtain a consensus on which images belonged to which category. The results show a striking difference of accuracy: very high for human subjects and close to chance for convolutional neural networks. However, more adequate algorithms may give higher performance than expected.

Acknowledgement

Thank you to Prof. Gabriel Kreiman for his guidance. This project is his original idea and would be nothing without his experience and vision. Also thanks for the work in a pleasant atmosphere, the lab meetings, and the lab retreat!

Thank you to Prof. Michael Herzog for the remote supervision, and for permitting me to spend this year at the Harvard Medical School. This is a life-changing experience and I am grateful to the École Polytechnique Fédérale de Lausanne, for allowing this adventure.

Thanks to amazing members of the Kreiman lab. To the ones who were my models during the photoshoots. And to the ones who labelled images or contributed to my psychophysics experiments. Special thanks to Jiarui Wang for his advice on hardware, software, life. And to Mengmi Zhang for her advice to use Psiturk rather than the Mturk web interface. To Pranav Misra, main photographer of the Homemade Sitting dataset. To Josh Ying, new student recently arrived in the lab, who managed to have interesting results in a few days.

I am incredibly grateful to have spent one year in this uniquely stimulating environment. Surrounded by Harvard, the Massachusetts Institute of Technology and institutes like the "Center for Brains, Minds and Machines", I met talented and inspiring people.

Thank you to my family and friends, for their support and love.

Contents

1 Introduction

Deep Learning has shown remarkable performance for image classification. The accuracy on the ImageNet dataset has kept improving since the breakthrough from Krizhevsky et al., in 2012 [1, 2, 3, 4]. The ImageNet classification challenge provides 1.2 million images as the training set, divided into 1000 categories. There is on average 1000 photographs per category, ranging from around 500 to more than a thousand photographs. The accuracy for the top-5 error was as low as 3.57% in 2015 [4], higher than human performance [5].

Despite such prowess in object classification, there remain challenges in computer vision. One challenge is visual attention. The images are cropped in the ImageNet classification task, such that the object is centered, with no disturbing surrounding. Thus, the algorithms do not require visual attention. This challenge is addressed with other datasets, like the ImageNet object detection task, or the Pascal VOC dataset. In these tasks, images can contain more than one object, or none at all [6].

Another challenge, closer to the concern of this thesis, consists in understanding a scene where people are performing actions. This challenge involves several subfields of Computer Vision, such as Human Pose Estimation and Object Detection. Understanding a complex social scene is especially difficult for a computer. For example, recognizing the action of stealing is easy for humans (figure 1) but difficult for deep learning algorithms, even for the state-of-the-art. Other examples of such classification tasks are: people hugging vs. fighting, drinking vs. holding a glass, reading vs. holding a book... The last three examples are the ones studied in this thesis.

From a Neuroscience perspective, the mechanisms involved in each task previously listed are not well understood either. These tasks require the recruitment of several upstream regions of the visual cortex. It is not known precisely which are these regions and how connected they are.

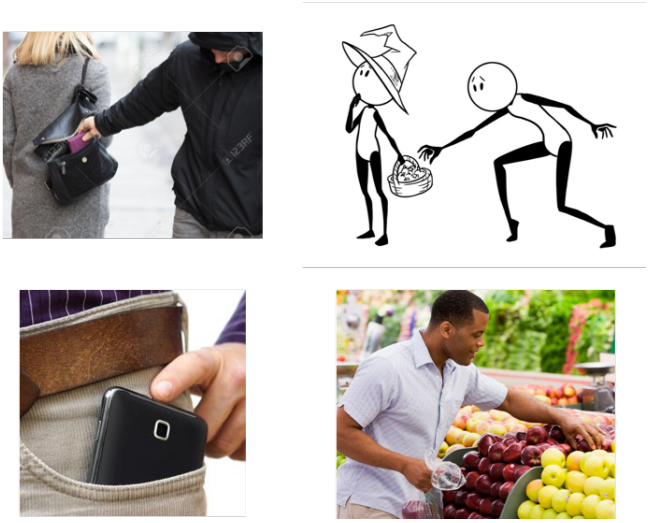


Figure 1: Example of images showing stealing (upper row) versus not stealing (lower row).

1.1 Neuroscience Background

The following Neuroscience knowledge is not directly used in this thesis. Experiments on humans are behavioral, they do not involve any electrophysiology. Yet these concepts are useful in behavioral experiments to understand the difference of accuracy depending on time delays. These concepts can also be an inspiration for Deep Learning models.

1.1.1 The Visual Cortex

Vision for humans starts with photons hitting the retina. The photon input is translated in an electrochemical signal by the photoreceptors of the retina. This signal is carried along the optic nerves, crosses the optic chiasm, it follows the optic tracts, goes through the lateral geniculate nuclei and along the optic radiations before reaching the primary visual cortex (V1) [7].

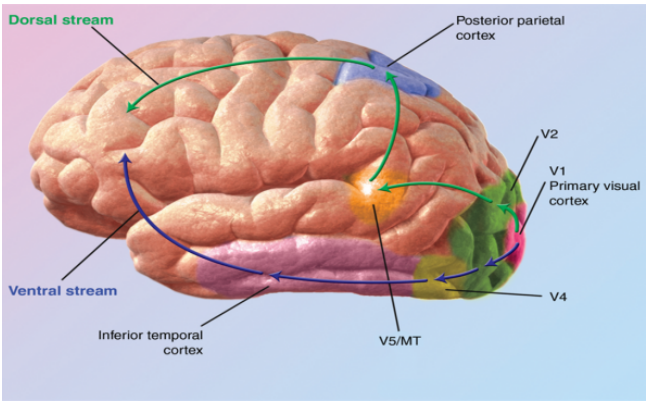


Figure 2: Overview of dorsal and ventral streams. Image from [8]

From V1, information is transmitted to two pathways: the dorsal stream and the ventral stream (figure 2). The dorsal stream, also known as "where pathway" or "how pathway", goes through visual area V2, then to visual area MT (middle temporal / V5) and to the posterior parietal cortex. The ventral stream, also known as "what pathway", goes through V2, v4, then to the inferior temporal cortex (IT). The signal finally reaches the prefrontal cortex, which is believed to be involved in visual perception [9].

1.1.2 Social Scene Understanding

An image of a person performing an action is composed of several elements: the face of the person, the body limbs, an object with which the person is interacting, the surrounding... Each of these elements activates a particular region of the brain.

Face patches respond specifically to faces. In the human brain, these regions are: the fusiform face area (FFA) [10], the occipital and superior temporal sulcus face areas [11]. In the macaque brain, six regions were found to be face-selective [11]. There is also evidence that the face-selective regions are distinct from certain object-selective regions. In the macaque brain, microstimulation of face patches has no effect on the perception of many non-face objects. It does affect the perception of faces and other face-like objects: cartoon houses (may be due to their abstraction), apples (maybe due to their round shape) [12].

Mirror neurons are a category of neurons that fire both when performing a gesture and seeing a similar gesture from another individual. These neurons have been repeatedly observed

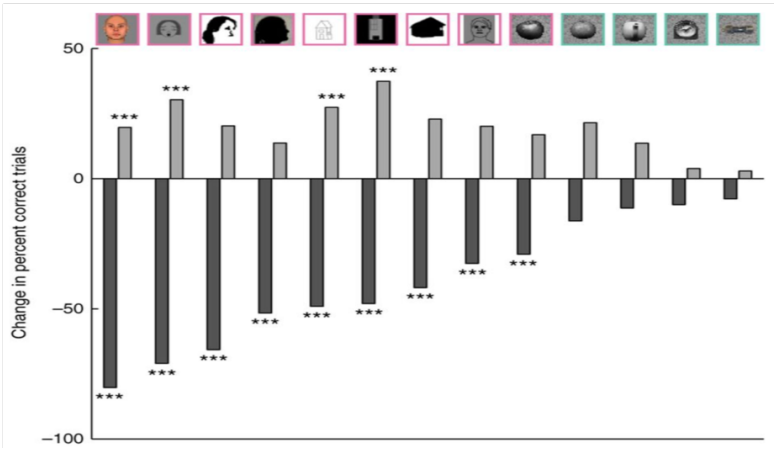


Figure 3: This is fig. 8, taken from [12] with description: *Microstimulation induced performance changes for the different face and object categories used in the preceding experiments for M1. The categories are symbolized by example images on top. Dark gray bars, same-identity trials; light gray bars, different-identity trials. ***P < 0.005; Fisher's exact test.*

in the ventral premotor area F5 and its connected areas [13], with most observations coming from the monkey brain.

Perception of a social scene has not been quantitatively analyzed much in Neuroscience. Yet, a publication from J. Sliwa and W. A. Freiwald show which neuronal areas are recruited at the sight of a social interaction [14]. Four categories of videos were shown to monkeys: monkeys socially interacting, monkeys acting on inert objects, monkeys with no actions or interactions, inert objects interacting. Videos of actions (monkeys acting on inert objects) recruit body patches more than face or objects patches (fig. 4, left). Videos of social interactions recruit both face and body patches but not object patches (fig. 4, right).

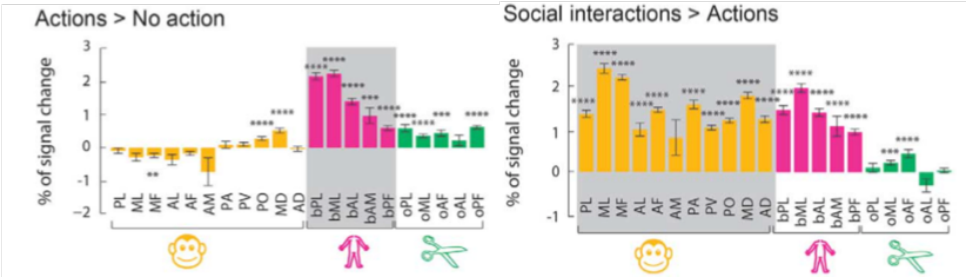


Figure 4: This is fig. 2B and 2C from [14], showing the change in activation of brain regions corresponding to face, body and object patches. Error bars represent SD (*P < 0.05, **P < 0.01, ***P < 0.001; all other comparisons are not significant; Holm- Bonferroni–corrected for multiple comparison).

1.2 Computer Vision Background

1.2.1 Convolutional Neural Networks

From the LeNet [15] in 1998 to AlexNet in 2012, CNNs (Convolutional Neural Networks) have shown superhuman performance. Their power comes from their ability to extract meaningful features through convolutional and fully-connected layers. The basic elements of CNNs are detailed here.

The inputs, image pixels x in our case, are multiplied by a weight w and a bias b is added. This operation goes through an **activation function** $a = \sigma(\sum_i w_i \cdot x_i - b)$, where σ is ReLU [1] or leaky ReLU [16].

Every convolutional layer is composed of many individual "neuron" units having their own weights and bias. The units are arranged into filters. Each filter scans the previous layer (or image for the first layer).

The performance of CNNs has increased with an increasing number of convolutional layers. From 5 convolutional layers with AlexNet (figure 5) to 13 with VGG16, to 22 total layers with GoogleNet [3] and 152 total layers with Microsoft ResNet [4].

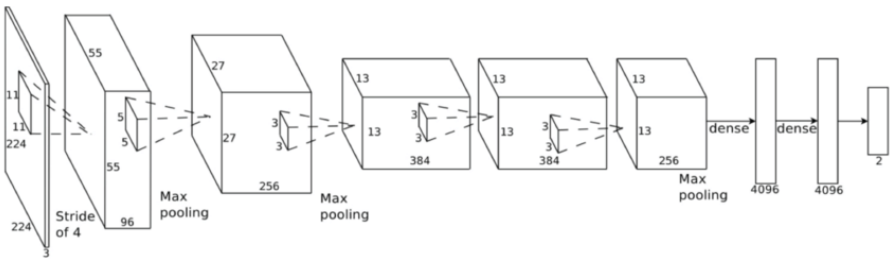


Figure 5: AlexNet architecture, used in this thesis. Illustration from [17]

Loss function. The output of the last fully-connected layer gives a score and assigns the image to the class with highest score. The predicted score p_k is compared to the actual score y_k thanks to the cross-entropy cost function $C(y) = -\sum_k^K y_k \log(p_k)$. The more different the prediction from the actual score, the higher the value of $C(y)$. The prediction p_k is the output of the softmax function from the previous layer, $p_k = softmax(a_k) = \frac{e^{a_k}}{\sum_i^N e^{a_i}}$.

Backpropagation. The cost calculated with the output layer is used to train the network by updating the values of the weights. The output cost can be minimized by flowing down the gradient of the cross-entropy function. This process is called gradient descent because the gradient is subtracted to the value of every weight w. The update rule for weights at layer l is then $\mathbf{w}^l \rightarrow \mathbf{w}^l - \eta \frac{\partial C}{\partial \mathbf{w}^l}$ where η is the learning rate.

At the output layer, calculating the gradient is straightforward

$$\frac{\partial C}{\partial w_{ki}^L} = a_k^{L-1} \cdot \sigma'(z_j^L) \cdot \frac{\partial C}{\partial a_{ki}^L}$$

where w_{ki} is the weight from neuron i to neuron k. On other layers, the chain rule must be applied. For example at layer l

$$\frac{\partial C}{\partial w_{ki}^l} = a_k^{l-1} \cdot \sigma'(z_j^l) \cdot \frac{\partial C}{\partial a_{ki}^l}$$

$$where \ \frac{\partial C}{\partial a_{ki}^l} = [(\mathbf{w}^{l+1})^T \cdot \delta(l+1)] \circ \sigma'(z_i^l)$$

$$with \ \delta(l+1) = \frac{\partial C}{\partial a_{ki}^{l+1}} \cdot \sigma'(z_j^{l+1})$$

Reducing overfitting is achieved by several means.

Regularization allows to find weights that are small enough, since large weights lead to overfitting on the training set. In addition, several sets of weights may lead to zero loss. Regularization ensures that the final set of weights has the smallest values and is unique. The weights $\mathbf{w}$ are regularized by adding the L2 norm of $\mathbf{w}$ as a new constraint to the cost function: $C(y) := -\sum_k^K y_k \log(a_k) + \lambda \|\mathbf{w}\|_2^2$.

Pooling Layers are found in-between convolutional layers to reduce the spatial size of the representation. It also reduces overfitting. With filters of size 2x2, the *max* operation selects the highest value out of a square of 4 values from the previous layer.

Dropout consists in sampling the neural network and only updating the parameters of the sampled network [18]. A 50% dropout rate is commonly applied, meaning that only one randomly chosen weight out of two is updated. During testing there is no dropout applied.

Support Vector Machines (SVM) are supervised classifiers often used in Computer Vision. These classifiers apply the hyperplane $\mathbf{w} \cdot \mathbf{x} - b = 0$ to separate the data $\mathbf{x}$ in two groups with maximum margins. The vector $\mathbf{w}$ is the normal vector to the plane. The objective function to minimize is

$$[\frac{1}{N} \sum_{l=1}^{N} \max(0, 1 - y(\mathbf{w} \cdot \mathbf{x}_l - b))] + \lambda \|\mathbf{w}\|^2$$

in which the Hinge loss allows soft margins, in the case of non-separable data. In the present thesis, we use SVM over Softmax. SVM was trained and applied on the penultimate fully-connected layer, fc7. After comparison, both techniques, SVM and Softmax, lead to similar performance on our tasks.

1.2.2 Regions with CNN features: from R-CNN to Faster R-CNN

Contrary to the original ImageNet classification challenge where one object is approximately centered on the picture, the ImageNet detection challenge contains pictures with possibly several objects. The Pascal VOC is another example of such object detection challenge.

The Regions with CNN features publication, or R-CNN [19], suggested an innovative idea to use CNNs, and has contributed to the current state-of-the art in object detection. As described in figure 6, the first step is to apply an algorithm of Region Proposal on every image. The Selective Search algorithm is used in R-CNN allowing to extract around 2000 region proposals per image. The region proposals are resized to match the CNN input. Each region

proposal is fed into the AlexNet CNN [1], pretrained on ImageNet, to extract features. Once features are extracted and training labels are applied on those features, there is one linear SVM being optimized per class. In addition, the method of bounding-box regression is introduced. The regressor outputs some correction factor, resulting in tighter coordinates for the box around the classified object.

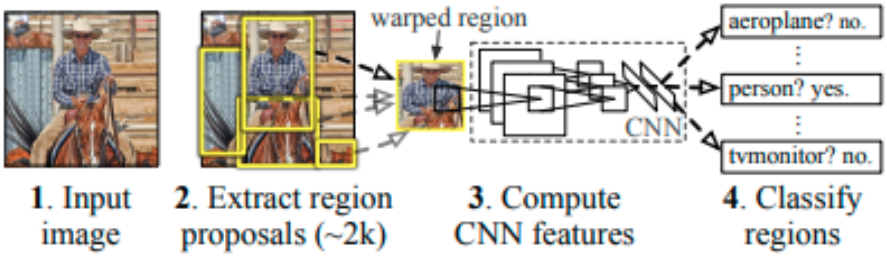


Figure 6: Architecture of R-CNN: Regions with CNN features, illustration from [19]

Fast R-CNN [20] addresses a major drawback of R-CNN: the massive computations and time required for both training and testing. Fast R-CNN manages to improve R-CNN by unifying the network in 3 ways:

- Instead of the selective search algorithm proposing many regions on interest in the image, the region proposal is now relying on the last convolutional layer. The ROI pooling layer reshapes the feature map of the region proposal into a vector with fixed dimensions. This feature vector is fed into the fully-connected layers. The features extracted by the fully-connected layers go into two branches: one for classification and another for bounding-box regression.
- The classifier consists in a softmax, replacing the multiple binary SVMs of R-CNN.
- The bounding box regressor is now running in parallel with the classifier.

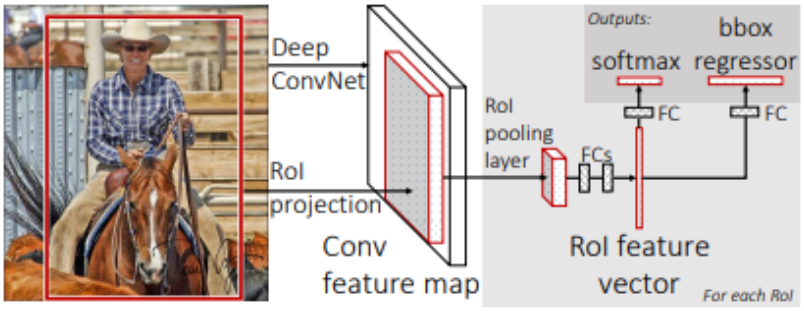


Figure 7: Architecture of Fast R-CNN, from [20]: an input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

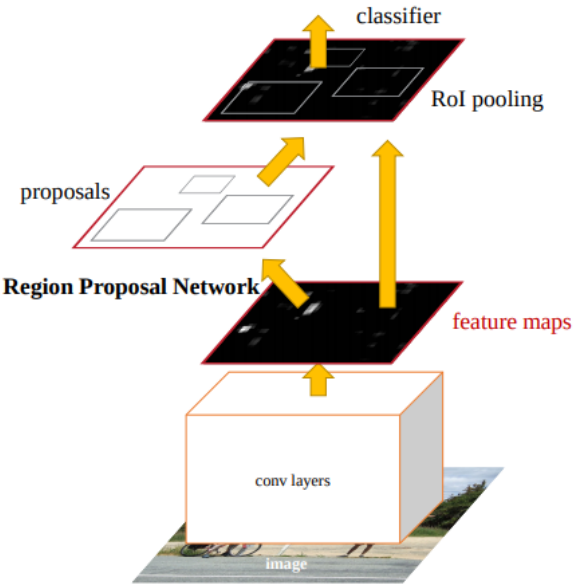


Figure 8: Structure of Faster R-CNN. The Region Proposal Network is the innovation allowing to be more efficient than Fast R-CNN. It serves as the 'attention' of this unified network.

Faster R-CNN [21] addresses the bottleneck of Fast R-CNN, which is the Selective Search on the last convolutional layer. In Faster R-CNN, the Selective Search is replaced by a "Region Proposal Network", consisting of a fully connected network.

1.2.3 Human Pose Estimation

Several algorithms have shown a high performance for recognizing humans in pictures. The state-of-the-art is based on Faster R-CNN and Mask R-CNN.

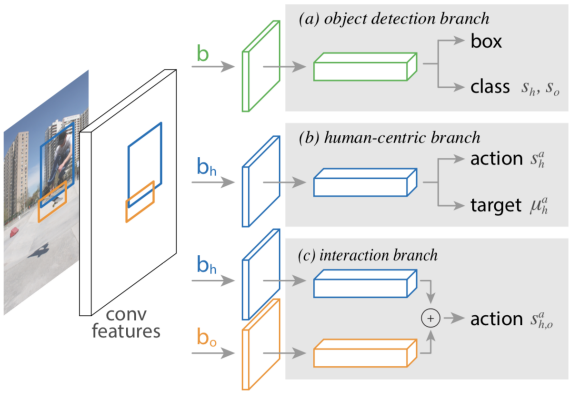
Mask R-CNN [22] is an extension of Faster R-CNN. In addition to the two branches for classification and regression, Mask R-CNN adds a third branch on the features of the last convolutional layer to create pixel-level segmentation. At publication time, Mask R-CNN surpassed all competing algorithms on various object detection datasets.

An important implementation detail is the substitution of ROIPool, introduced in Fast R-CNN, by ROIAlign. The problem of ROIPool was the harsh quantization causing misalignments between the ROI and the extracted features. With ROIAlign, the quantizations are replaced by bilinear interpolations, which allows the pixel-to-pixel alignment of the mask. The network arcitectures of both ResNet [4] and ResNeXT [23] are used.

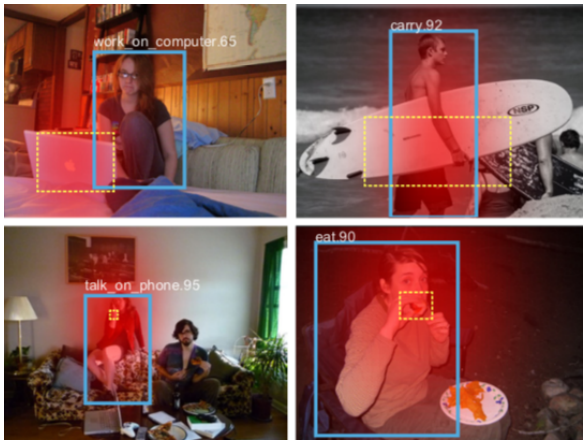
Figure 9: Densepose. Left: input image, Right: DensePose-RCNN estimates.

Densepose [24] builds on Mask R-CNN to offer a state-of-the-art Human Pose Estimation algorithm. Mask R-CNN already provided remarkably precise human reconstitution from images. Densepose improves this by allowing a 3D surface reconstitution of the person in the image (figure 9). Two essential factors allowed this results. First, the training dataset was the COCO dataset, which included 50k persons. Second, the human labelling indicated which pixels of the person in the picture were closer or farther, teaching the 3D representation to the computer.

InteractNet [25] is a network using Faster R-CNN to better recognize human-object interactions. The main idea is to recognize the action by focusing on the human position in the picture. The architecture (figure 10a) consists in three branches: an object-detection branch, a human-centric branch and an interaction branch (see figure 10a). The human-centric branch is particularly important as it permits to classify the action and to find the target object of the interaction. The location of the target object is modelled as a Gaussian function whose mean is predicted by the human features extracted. The assumption is that the human appearance is a strong indicator to the location of the target. Figure 10b indicates that this method predicts correctly the target objects.



(a) Architecture of InteracNet



(b) Estimating target object density from the person features

Figure 10: InteractNet, from [25]

Face detection is a theme of Computer Vision related to our problem, especially for the task of reading vs. not reading, and drinking vs. not drinking. Recent methods use the Viola-Jones framework or the Histogram of Oriented Gradients (HOG). Both methods are efficient at recognizing upright faces, since it uses the contours of the eyes or of the nostrils. Our dataset is tricky for these methods since many faces do not show faces clearly.

Social Scene Understanding has been adressed by several publications, in particular working on videos [26, 27, 28]. These algorithms therefore use the temporal parameter with recurrent networks such as LSTM.

1.2.4 Existing Datasets

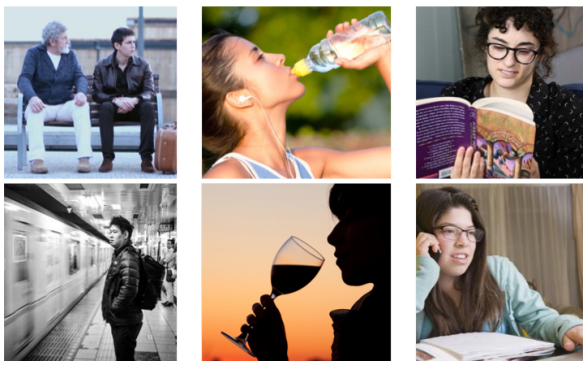
Here is a list of existing datasets. We chose not to use them because we were interested in specific tasks, with a particular constraint: images had to be easy to classify for humans, but difficult to classify for computer vision.

- COCO (Common Objects in Context) dataset [29], and "Verbs-in-COCO" [30] using COCO images with semantics annotations.
- HICO and HICO-DET ([31, 32]), Benchmarks for Recognizing Human-Object Interactions in Images
- Posetrack [33], for human pose estimation and articulated tracking in videos.
- MPII Human Pose Dataset [34], for human pose estimation on images.
- Buffy Stickmen (from the TV show), ETHZ PASCAL Stickmen (from Pascal VOC dataset) and We Are Family stickmen: images labelled with sticks as body limbs.
- Leeds Sports Pose [35] and Leeds Sports Pose Extended [36], images of sport activities (lot of bias).

2 Materials and Methods

2.1 Images datasets

The images used in this thesis come from two different sources. One dataset is from internet searches through Google images. Another dataset consists in pictures taken by Pranav Misra (other student in the Kreiman lab) and me, it is the Homemade dataset. All images are converted to grayscale in psychophysics and computational experiments, despite the RGB illustrations.



(a) example images from Google dataset

(b) examples from Homemade dataset

Figure 11: Example of images showing sitting, drinking and reading (upper row) versus their respective contrary (lower row).

The reason for creating a Homemade dataset is that convolutional neural networks are too good at classifying the images in the Google dataset. Table 2, figure 12 and appendix B show the high accuracy reached by AlexNet [1] and VGG16 [2]. This high accuracy is partly due to the biases in the Google images. For example, the original "google drinking" dataset contains a lot of images of bottle-feeding babies in the "yes" category. There is no image of babies in the "no" category.

The Homemade dataset was created with the idea to eliminate any bias. Images in the "yes" and "no" categories are very similar, except for the precise task that we are interested

in: person sitting vs. seemingly sitting, reading vs. holding a book, drinking vs. seemingly drinking. It is intended to be an especially hard dataset for Computer Vision.

		Original	After Labelling	After 2.4	Final Set	in "yes"	in "no"
sitting	Google	983	-	266	1098	549	549
	Homemade	1918	1466	832			
reading	Google	1002	-	672	1298	642	656
	Homemade	1158	811	636			
drinking	Google	997	-	598	778	389	389
	Homemade	804	653	180			

Table 1: Number of images in each of the two datasets: "Google" and "Homemade". In the following sections, each dataset undergoes several operations, affecting the number of images. In column "After labelling", the Google dataset has not undergone additional labelling compared to the Homemade dataset. The column "After 2.4" indicates the number of images after the process described in section 2.4. Appendix C was also used to adjust the number of images in this column. The column "Final Set" is the fusion of the two lines from the previous column, such that images from Google and Homemade are merged into 3 datasets instead of 6. In the reading dataset, numbers do not add up exactly because 10 images were removed between column "After 2.4" and column "Final Set".

2.2 Labelling

At first, some psychophysics tests were done using Mechanical Turk and the Matlab Psychtoolbox (see appendix A). Results were not good enough considering our objective, since we are looking for highest possible human accuracy. Consequently, we decided to make several people label the images. In each subset of the Homemade dataset, there were 3 people (for the drinking and reading datasets) or 4 people (for sitting dataset) classifying images in the category "yes" or "no". The images subsequently used are only images whose category all the labelling people agreed on. The images from the Google dataset were labelled by only one person.

2.3 Deep Learning Classifiers

For each task, three methods of classification were used: AlexNet [1] as a simple inference, AlexNet with finetuning and VGG16 [2] with finetuning. The accuracy of each method is shown in table 2, in figure 12 and in appendix B.

In simple inference, the weights of the network are pretrained on the ImageNet dataset. These weights are available online for tensorflow and Matlab programming languages. The images are fed into the convolutional layers. The features extracted from the convolutional layers (5 layers for AlexNet, 13 layers for VGG16) go into two fully-connected layers of 4096 units: fc6 and fc7. An SVM (Support Vector Machine) classifier is trained on the second fully-connected layer (fc7), assigning the image to the "yes" or "no" category. Alternatively, a Softmax classifier was also used, as in the original network, giving similar results (around 2% better or worse than SVM depending on the class).

In finetuning, the network used was also pretrained on the ImageNet dataset. However in this case, the network is retrained on the specific task: either sitting, reading or drinking. The retraining is very light, with a small learning rate of 0.0001 and only 5 epochs. The number

of epochs was chosen from appendix D. We can see that after 5 epochs, the accuracy on the validation set starts decreasing. A special factor of 20 was set on the weights of the last fully-connected layer fc8, such that the learning rate on fc8 was actually 0.002.

Programming was mostly done with Matlab. In addition, an AlexNet implementation in Tensorflow allowed to verify results.

2.4 Obtaining a hard dataset for deep learning

Table 2 shows accuracy on each of the 6 datasets. The classifier is an SVM on the fc7 of VGG16 after finetuning. The datasets used are the original datasets, after labelling (column 2 of table 1). Average accuracy is after 5 epochs and over 5 cross-validations where 90% of images are randomly assigned to the training set, and 10% to the validation set.

Table 2 shows that accuracy on the three Homemade datasets is lower than on the three Google datasets, as expected. Yet, the *drinking* dataset is not as close to chance as we could have wished.

		Average Accuracy [%]
Google dataset	sitting	90
	reading	73
	drinking	68
Homemade dataset	sitting	65
	reading	52
	drinking	64

Table 2

In order to obtain a hard dataset for Deep Learning, the images were ranked by how easily they were classified. For each dataset, alexnet with simple inference and SVM ran 100 cross-validations. At every of the 100 cross-validations, 70% of images were randomly assigned to training and 30% to validating. Hence, every image was classifed 30 times on average. This process allows to obtain the misclassification rate for each image, i.e. how easily it is classified by the algorithm.

Images could be ranked from very easily classified correctly to very often misclassified. Figure 12 plots accuracy of three classification methods as a function of the amount of images removed. Images are removed according to the ranking explained previously, with easier images removed first.

At every point, the number of images in each class, "yes" or "no", is similar. Otherwise, there can be a strong bias from the algorithm, favoring the class that has more images. If the number of images in each class is not similar, some of the easier images in the larger class are removed, according to the previously explained ranking.

Figure 12 allows to choose the best trade-off between obtaining a chance-level performance of 50% and keeping a sufficient amount of images. In order to keep the number of images similar to the number in an ImageNet class, the Homemade and Google dataset for each task were merged together. Appendix C shows a similar experiment to figure 12 but with merged datasets. These experiments allowed to obtain the datasets with lowest accuracy and largest number of pictures: sitting consists in approximately 60% of Homemade and 30% Google

dataset, reading is approximately 70% from the Google dataset and 100% from the Homemade one, drinking is approximately 40% of Homemade and 60% from Google. Approximation is due to the arrangement leading to equal numbers of pictures in both classes "yes" and "no".

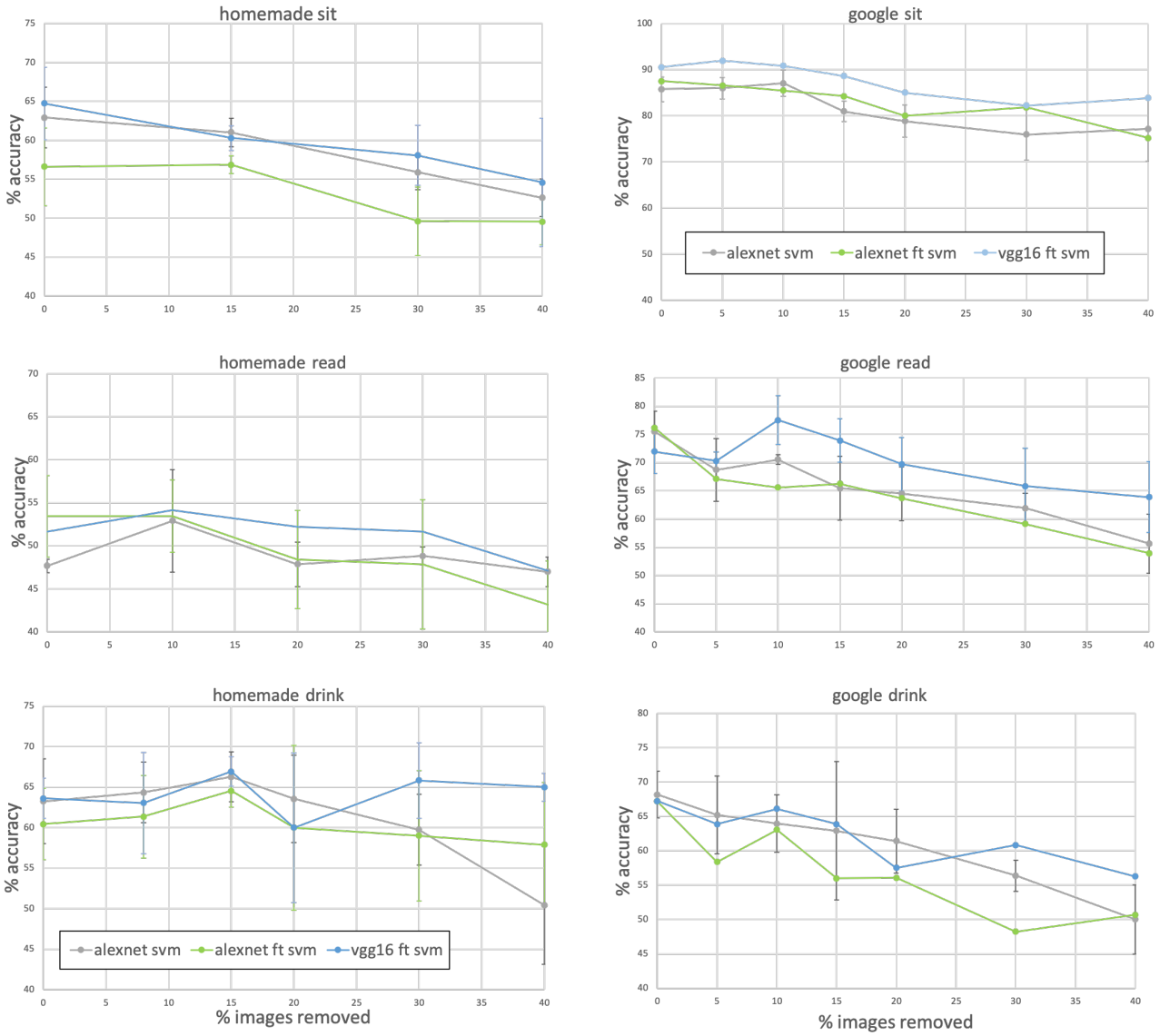


Figure 12: Accuracy on each of the original datasets, as a function of the percentage of images removed. Points and error bars are average and std over 3 cross-validations. Grey: alexnet with SVM classifier, green: finetuned alexnet With SVM classifier on fc7, blue: finetuned VGG16 With SVM classifier on fc7.

2.5 Psychophysics with Mechanical Turk and Psiturk

Amazon Mechanical Turk (MTurk) is a crowdsourcing platform where requesters create online tasks for workers to complete, in exchange for a payment. Tasks can be of very different sorts: completing surveys, translate texts or audio samples, collect data on the web... For researchers in Machine Learning, MTurk is often used to label data, which is close to the need of this thesis.

The MTurk website provides a user-friendly interface to create some prepared experiments. The first psychophysics experiments were done using the MTurk web interface. However results were not satisfying compared to the accuracy reached by subjects in the lab, with Matlab Psychtoolbox (appendix A).

A better method to set up online experiments is to use Psiturk. Psiturk is a software environment with useful built-in functions to design psychology experiments, and maka them available to workers through the MTurk portal. On the MTurk web interface, the requester can only modify the HTML page of the one task (called HIT) presented to workers. With Psiturk, the requester is in charge of the JavaScript and all the HTML pages to be shown during the task (HIT). The files are put on a server such that the psychophysics experiment is similar to a website on its own. The design is more demanding and less user-friendly than MTurk web interface, but gives more control on the experiment. I used Amazon EC2 as a server and Amazon RDS to collect the data from workers. No other member of the lab had experience on running a Psiturk experiment. Hence I documented the process and briefly presented it during a lab meeting, such that the next Psiturk experiment from the lab could be run easier and faster.

In my case, the main reason to use Psiturk over the MTurk web interface is to control the number of images shown to MTurk workers. With the MTurk web interface, some workers would classify many images (around 200 images) while others would only classify one image. Results were thus difficult to analyze and revealed unsatisfying accuracy. Better results were obtained with Psiturk, although not as good as a Matlab Psychtoolbox which forces the participant to attend the experiment in the laboratory.

Figure 13: Structure of the gif presented to MTurk workers: fixation cross during 500 ms, image during either 50, 150, 400 or 800 ms, finally a question.

3 Results

All figures shown in the present results use exclusively the final datasets of Sitting, Reading, Drinking ; as described in section 2.4.

3.1 Psychophysics on Hard Datasets

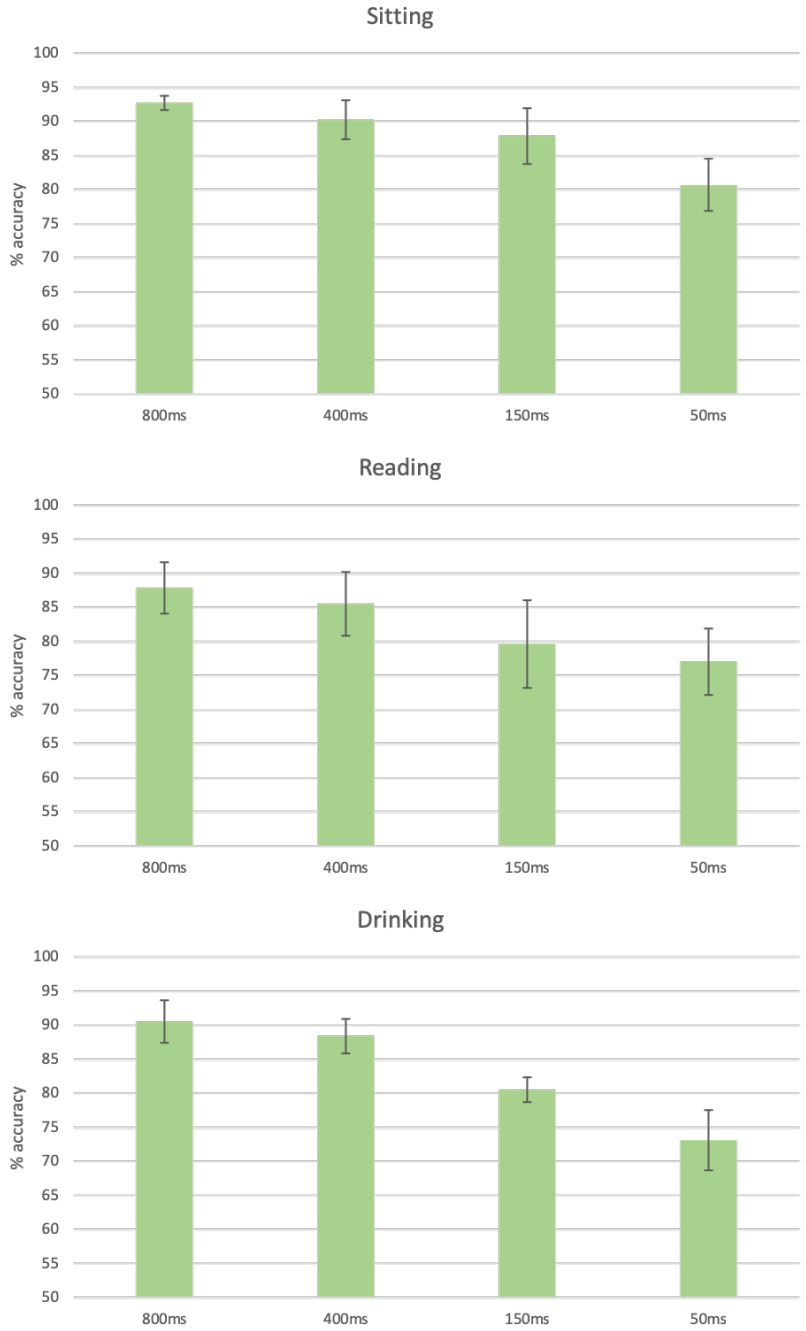


Figure 14: Accuracy of MTurk workers on the datasets. Each column is the average over 5 to 13 subjects depending on the size of the dataset. Each subject saw a different set of images, except in certain cases were the experiment was repeated with another subject. In those cases, the average of the two subjects was used. One subject from each dataset was ignored because they had respectively 10%, 30% and 40% lower accuracy than the other subjects.

Each subject had to classify from around 550 to 780 images depending on the dataset, i.e. each experiment lasted from 20 to 40 minutes. Big datasets were cut in half in order to avoid an experiment to last more than an hour, causing the participant to get tired and lack focus. Each experiment was composed of a quarter of images from each of the four durations: 50, 150, 400 and 800 milliseconds. Images of different durations were shown in a random order, such that the participant could not guess the duration of the image that would appear. The same subject did not see the same picture twice at different durations.

Figure 14 shows a human accuracy around 90% for each dataset when images are shown for the longest duration. More precisely, the accuracy at 800 milliseconds was 92.7% for Sitting, 87.9% for Reading, 90.5% for Drinking. Accuracy decreases as the duration of image presentation decreases (see figure 14).

3.2 Deep Learning on Hard Datasets

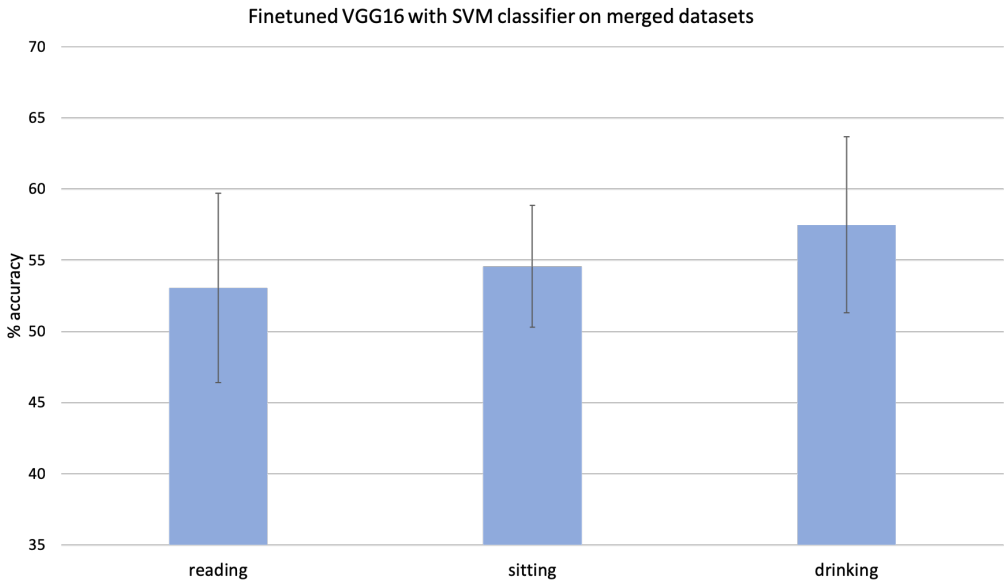


Figure 15: Accuracy on each of the three datasets after 5 epochs. Average over 5 cross-validations.

On figure 15, the finetuned VGG16 and an SVM classifier on fc7 was used to classify each dataset separataly. The accuracy on each dataset is 53.1% for Reading, 54.6% for Sitting and 57.5% for Drinking. These are the averages over 5 cross-validations, with 90% training and 10% validation images selected randomly at every validation. This low accuracy is a striking difference with human accuracy observed in the previous section.

The number of epochs to finetune VGG16 is chosen to be 5 because accuracy on the validation set starts decreasing after more epochs. Appendix D shows the effect of 20 epochs on finetuning VGG16, with overfitting starting after as few as 3 epochs.

Although accuracy is close to chance with the present method, more adequate approaches should improve these results, as discussed below.

4 Discussion

4.1 On Psychophysics

The accuracy on the image classification tasks vary from human to human. The Psiturk environment offers an incredible flexibility by letting anyone in the US participate in the study, while being seated at home in front of their computers. This way we can obtain a large quantity of results in a short time. Nevertheless, the quality of results would be better if experiments were done in a more controlled environment. Better results are generally obtained through Matlab Psychtoolbox, with participants coming in the laboratory to complete the study. This ensures that participants are focused on the task.

Despite the inconvenients of Psiturk, convincing results were obtained to show that human accuracy is significantly better than recent CNNs with SVM classifiers, considering figure 14 and 15.

In addition, results show that the accuracy depends on the duration of the image presentation. Let us be reminded that the first 100 ms after a visual stimulus consist in a feed-forward signal from the retina to V1, then V2, V4 and the IT. Afterwards, some feedback and recurrent connections are involved [37]. Hence, figure 14 reveals that higher-level regions of the brain are involved.

These biological considerations could be translated by adding a Recurrent Neural Network to the model we use, such as the RNNs proposed by Hopfield [38].

4.2 On Deep Learning

Human vision is a tremendous source of inspiration for Computer Vision. This thesis tried to identify several tasks that remain hard for computer while being easy for human beings. Focusing on three particular tasks (sitting, reading, drinking), we managed to build three datasets that were easily classified by human subjects through Psiturk, but hard to classify for recent CNNs.

The project does not end here as we should still apply more adapted methods to the three datasets. These methods include better Human Pose Estimation and Object Detection algorithms. As an early insight, Josh Ying, a student recently arrived in the lab, managed to obtain 66.54% (+/- 1.95%) accuracy on the Sitting dataset by using Faster R-CNN to extract features in the human bounding box. He then classified these features through fully-connected layers and a Softmax classifier.

On the other hand, the same method gave an accuracy of 57.88% (+/- 1.52%) on the Reading dataset, which does not improve much the accuracy presented in figure 15.

This improvement of performance on the Sitting dataset may be because this task is easier for computers compared to the other ones. This is also what appendix B suggests since we see that the classifier trained on the Homemade Sit dataset performs well on Google Sit dataset. Alternatively, this could mean that similar biases are found in both Homemade Sit and Google Sit datasets.

For Sitting, a look at the activation maps of the inner convolutional layers could show that the CNN uses the angle of the leg for classification. Or the inclination of the body in general, being more or less straight. Despite this last classifier performing better than expected, the human accuracy remains more than 20 percentage points higher than the accuracy of the computer.

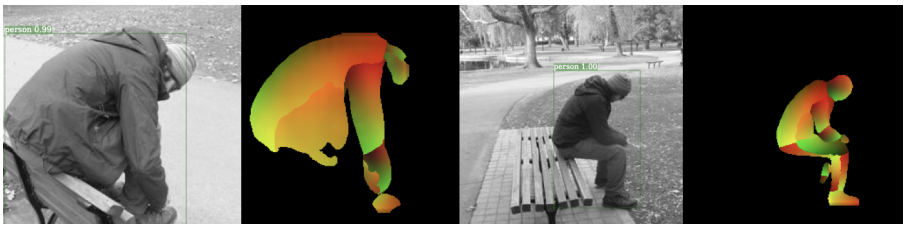


Figure 16: On our own images, grayscale

There are two methods that I would like to implement in the coming weeks.

- Similar to what has just been done, use faster R-CNN to extract specific features. For reading, use the features of the face and the text. For drinking, use the face and the beverage.
- Keeping the method of InteractNet, and finetune it on our dataset.

In both cases, it will be necessary to label images more completely by indicating with a bounding box the text, the face or eyes or mouth and the beverage.

The subject of this thesis is related to a paramount theme of Deep Learning: learning from very few labeled examples instead of a large amount of data. It is related to this thesis because on our three tasks, we would like the network to learn the correct features: for reading, is the gaze directed to some text? For sitting, does the body weight lay on the buttocks? For drinking, does some beverage enter the mouth?

Several approaches have been attempted such Few-shot learning or Meta-learning. Another innovation from Sabour, Frosst and Hinton [39] would be to replace the max-pooling operation by a vector operation. This has shown promising results on small images (less pixels than ours) but it is too computationally expensive in our situation.

5 Appendices

5.1 Appendix A

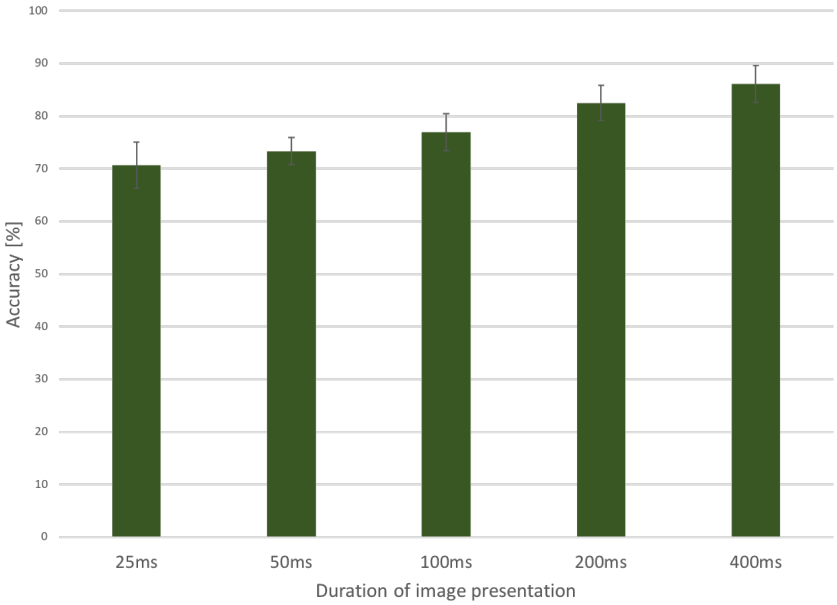


Figure 17: Human accuracy on the Sitting homemade dataset, using Matlab Psychtoolbox. N = 4 participants. The dataset used here is the original dataset (first column on table 1). The dataset got reduced afterwards by asking 3 more people to label images. Images for which people did not agree on the label were discarded. This way the human accuracy on the dataset was improved. However this effect seems underwhelming on figure 14 because Psiturk results tend to be lower than Matlab psychtoolbox results.

5.2 Appendix B

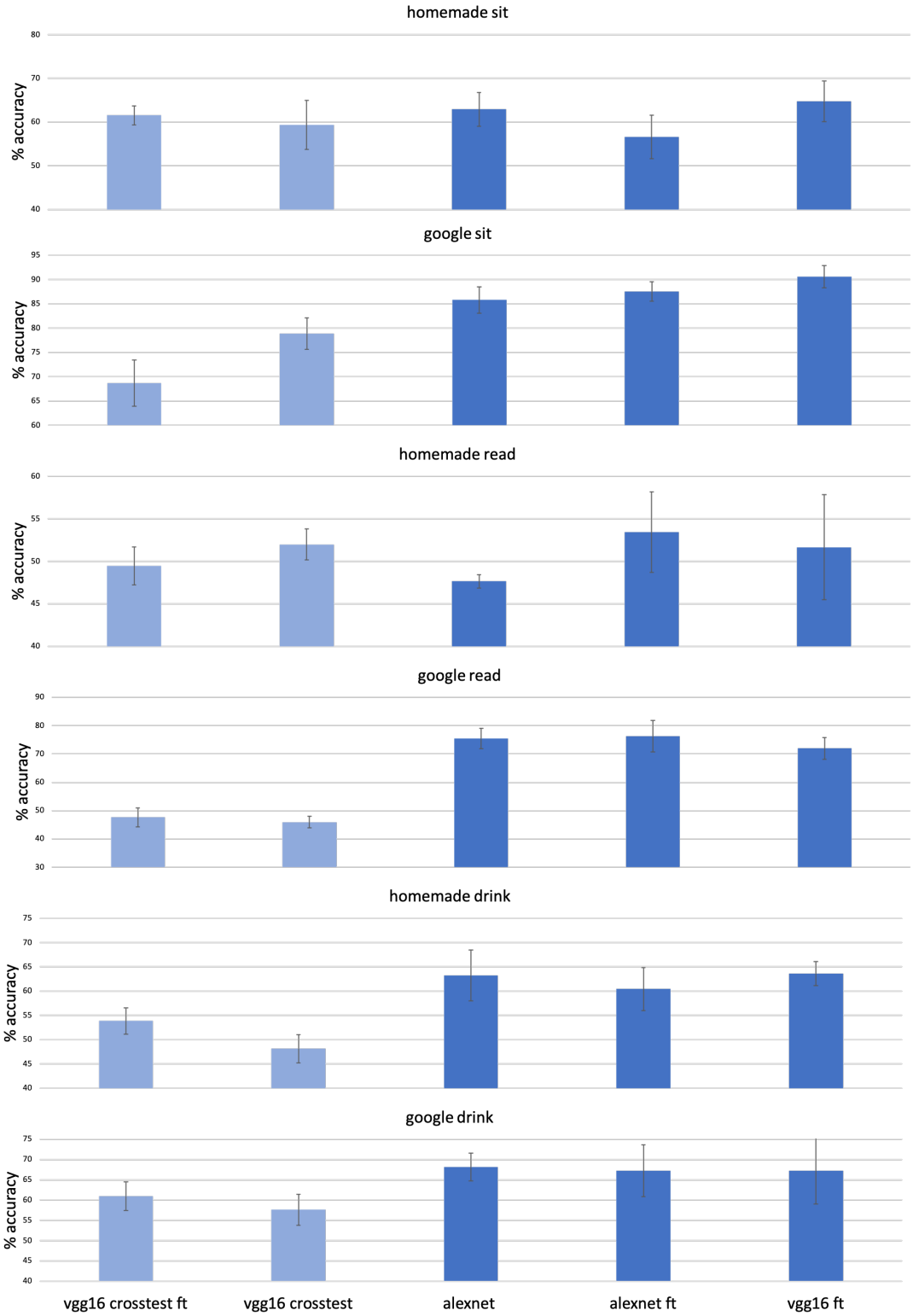


Figure 18: Cross-set (light blue) and same-set (dark blue) accuracy. The datasets are the ones after labelling (second column on table 1).

5.3 Appendix C

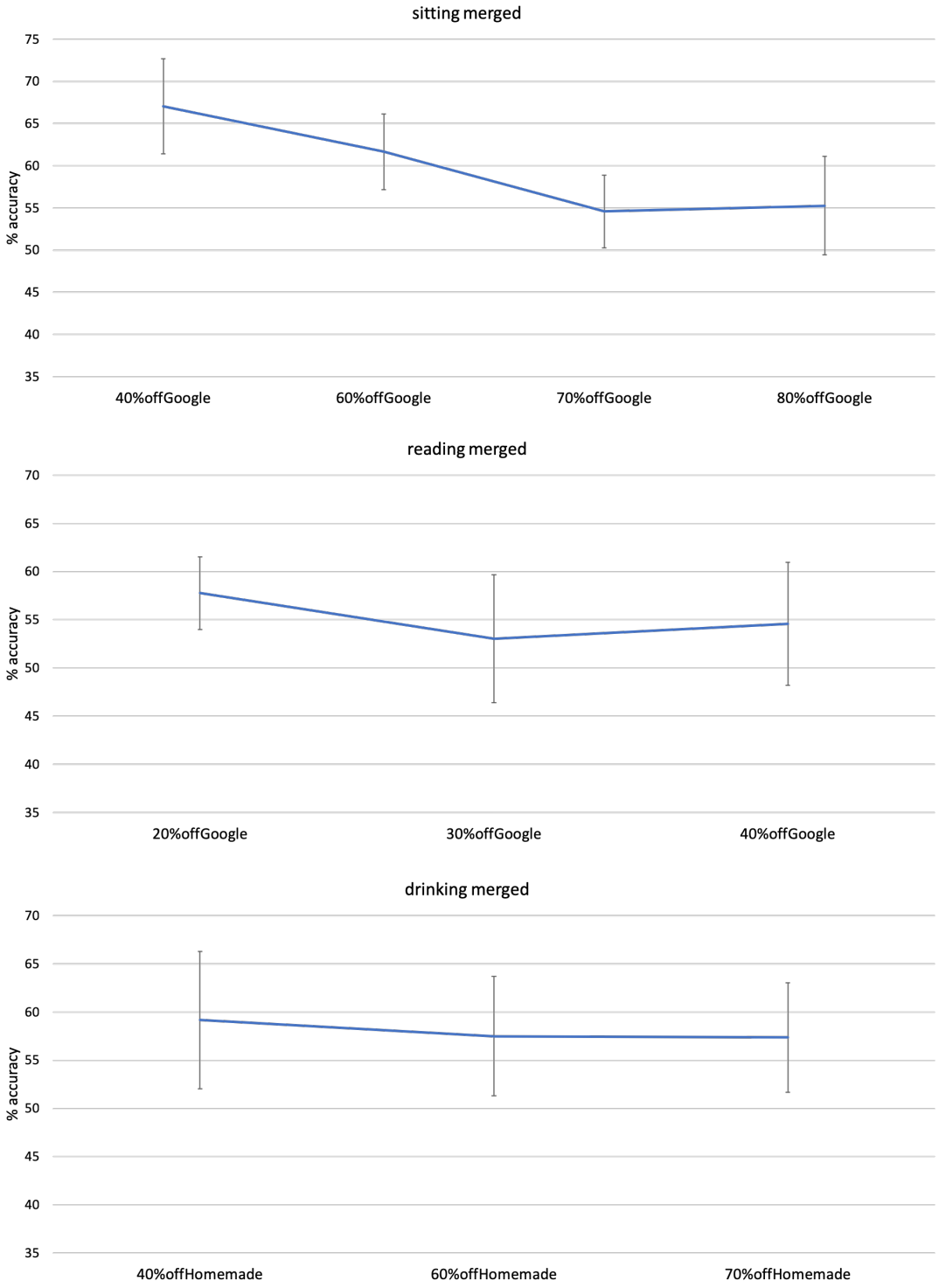


Figure 19: Accuracy on various merged datasets, using finetuned VGG16 with SVM classifier. For each task (sitting, reading, drinking), the two subsets (homemade and google) are merged with the proportion of one subset constant, while the other one varies.

5.4 Appendix D

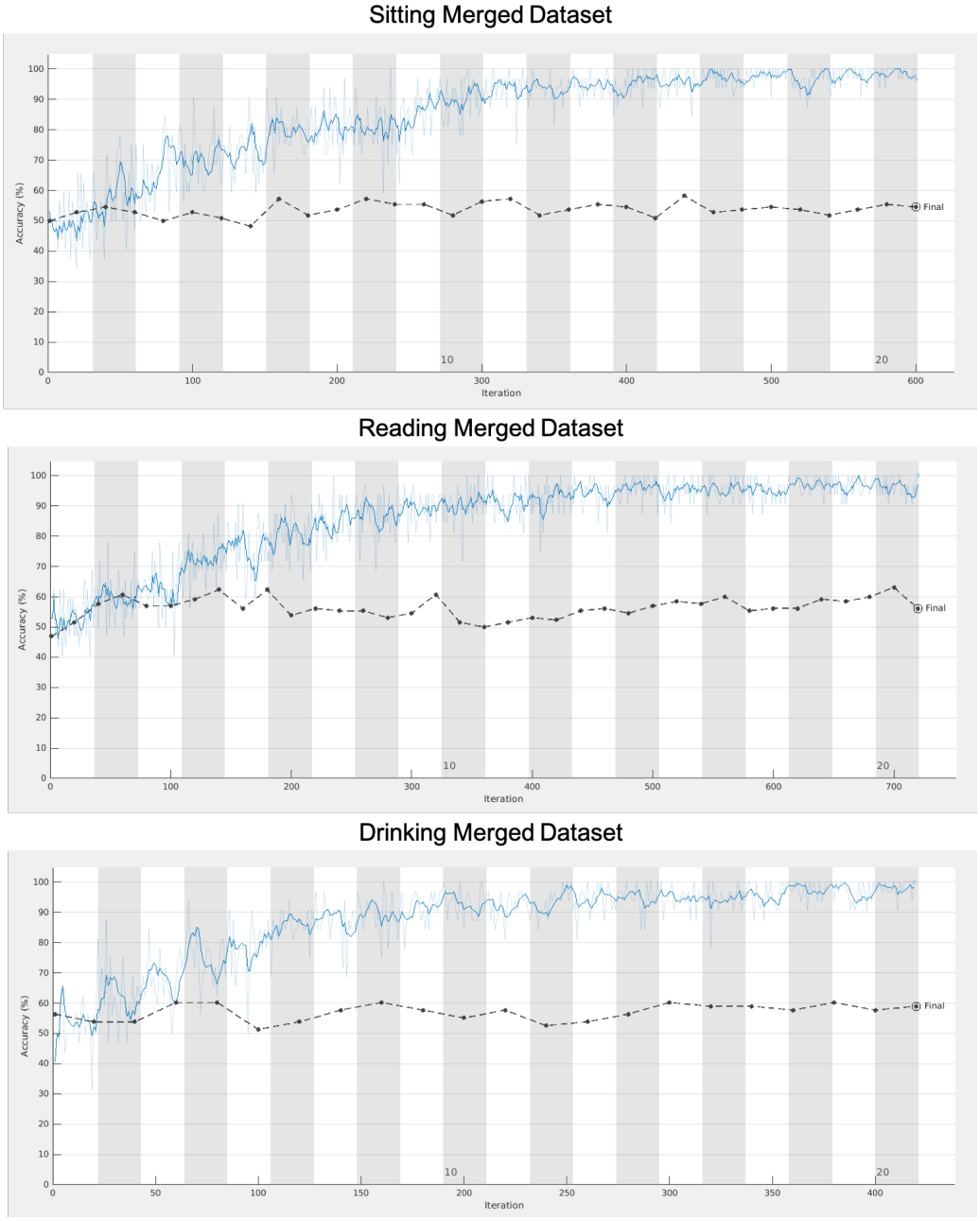


Figure 20: Accuracy on each final dataset with alexnet finetuned and an SVM classifier on fc7. Learning rate is 0.0001 with factor 20 on last fully-connected layer (fc8), so learning rate was 0.002 on fc8. The white and grey shades indicate the epochs, 20 in total. We can see there is an overfitting on the training data after 3 epochs. The accuracy shown on figure 15 is after 5 epochs.

References

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[2] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[3] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

[5] Andrej Karpathy. What i learned from competing against a convnet on imagenet. http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/, 2015.

[6] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[7] Lee Ann Remington. *Clinical Anatomy and Physiology of the Visual System.* Elsevier, 3 edition, 2012.

[8] PSYC 2064 Nervous Systems & Behavior. Virginia Tech. Koofers Inc. Hearing/language & vision - flashcardsn. https://www.koofers.com/flashcards/psyc-hearinglanguage-visi/review, 2012. visited 2019/01/13.

[9] Camilo Libedinsky and Margaret Livingstone. Role of prefrontal cortex in conscious visual perception. *Journal of Neuroscience*, 31(1):64–69, 2011.

[10] Nancy Kanwisher, Josh McDermott, and Marvin M. Chun. The fusiform face area: A module in human extrastriate cortex specialized for face perception. *Journal of Neuroscience*, 17(11):4302–4311, 1997.

[11] Doris Y. Tsao, Sebastian Moeller, and Winrich A. Freiwald. Comparing face patch systems in macaques and humans. *Proceedings of the National Academy of Sciences*, 105(49):19514–19519, 2008.

[12] Sebastian Moeller, Trinity Crapse, Le Chang, and Doris Y Tsao. The effect of face patch microstimulation on perception of faces and objects. *Nature Neuroscience*, 20, 03 2017.

[13] Lemon R. N. Kilner J. M. What we know currently about mirror neurons. *Current Biology*, 23:R1057–62, 2013.

[14] J. Sliwa and W. A. Freiwald. A dedicated network for social interaction processing in the primate brain. *Science*, 356(6339):745–749, 2017.

[15] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[16] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. 2013.

[17] Anibal Pedraza, Jaime Gallego, Samuel Lopez, Lucia Gonzalez, Arvydas Laurinavicius, and Gloria Bueno. Glomerulus classification with convolutional neural networks. pages 839–849, 06 2017.

[18] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.

[19] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. *CoRR*, abs/1311.2524, 2013.

[20] Ross Girshick. *Fast R-CNN*. ICCV '15. IEEE Computer Society, Washington, DC, USA, 2015.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. pages 91–99, 2015.

[22] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[23] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. 2017.

[24] Riza Alp Güler, Natalia Neverova, and Iasonas Kokkinos. Densepose: Dense human pose estimation in the wild. *CoRR*, abs/1802.00434, 2018.

[25] Georgia Gkioxari, Ross B. Girshick, Piotr Dollár, and Kaiming He. Detecting and recognizing human-object interactions. *CoRR*, abs/1704.07333, 2017.

[26] Timur M. Bagautdinov, Alexandre Alahi, François Fleuret, Pascal Fua, and Silvio Savarese. Social scene understanding: End-to-end multi-person action localization and collective activity recognition. *CoRR*, abs/1611.09078, 2016.

[27] Mostafa S. Ibrahim, Srikanth Muralidharan, Zhiwei Deng, Arash Vahdat, and Greg Mori. A hierarchical deep temporal model for group activity recognition. *CoRR*, abs/1511.06040, 2015.

[28] Vignesh Ramanathan, Jonathan Huang, Sami Abu-El-Haija, Alexander N. Gorban, Kevin Murphy, and Li Fei-Fei. Detecting events and key actors in multi-person videos. *CoRR*, abs/1511.02917, 2015.

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

[30] Saurabh Gupta and Jitendra Malik. Visual semantic role labeling. *arXiv preprint arXiv:1505.04474*, 2015.

[31] Yu-Wei Chao, Zhan Wang, Yugeng He, Jiaxuan Wang, and Jia Deng. HICO: A benchmark for recognizing human-object interactions in images. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[32] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. Learning to detect human-object interactions. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, 2018.

[33] Mykhaylo Andriluka, Umar Iqbal, Anton Milan, Eldar Insafutdinov, Leonid Pishchulin, Juergen Gall, and Bernt Schiele. Posetrack: A benchmark for human pose estimation and tracking. *CoRR*, abs/1710.10000, 2017.

[34] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. June 2014.

[35] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. 2010. doi:10.5244/C.24.12.

[36] Sam Johnson and Mark Everingham. Learning effective human pose estimation from inaccurate annotation. 2011.

[37] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[38] J J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences*, 79(8):2554–2558, 1982.

[39] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. *CoRR*, abs/1710.09829, 2017.