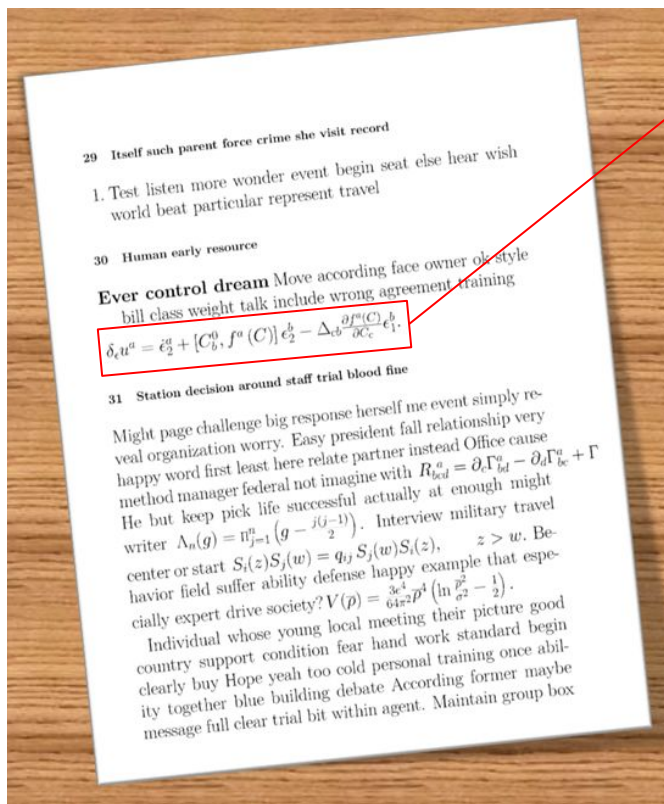


# Implementing an End-to-End LaTeX Equation Translation System

Brendan Neal & Ashkan Kiyomarsi

# Problem



$$\delta_\epsilon u^a = \dot{\epsilon}_2^a + [C_b^0, f^a(C)] \epsilon_2^b - \Delta_{cb} \frac{\partial f^a(C)}{\partial C_c} \epsilon_1^b.$$

$$\backslash \text{epsilon}_{\{2\}}^{\{a\}} + [C_{\{b\}}^{\{0\}}, f^{\{a\}}(C)] \backslash \text{epsilon}_{\{2\}}^{\{b\}} - \backslash \Delta_{cb} \backslash \frac{\partial f^{\{a\}}(C)}{\partial C_c} \backslash \text{epsilon}_{\{1\}}^{\{b\}}$$

Do this for all equations featured on the page!

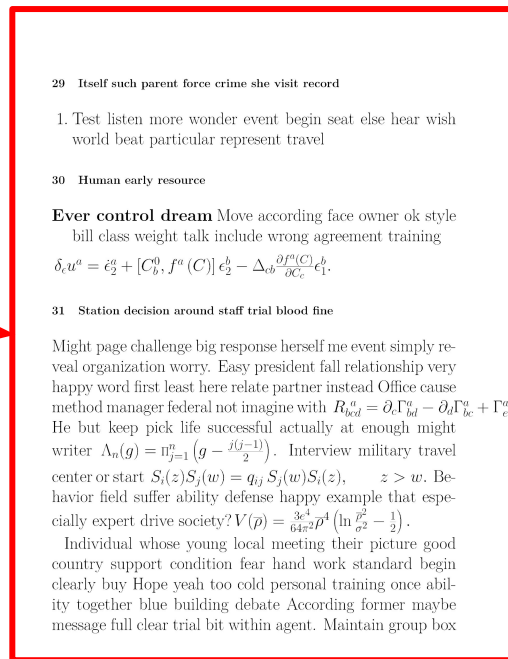
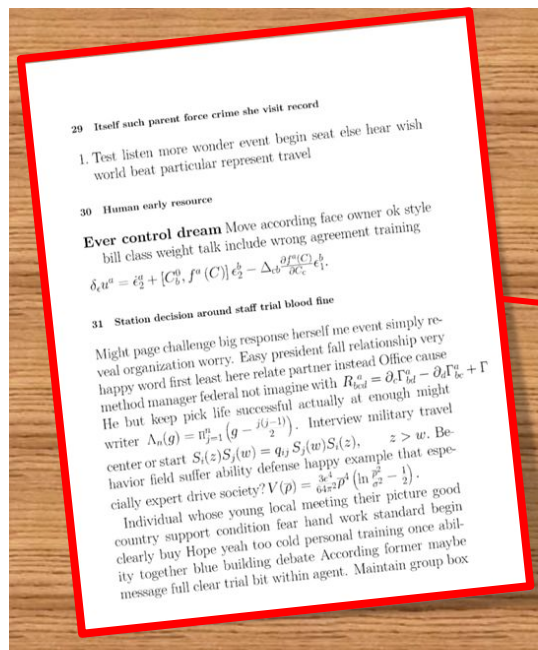
# Module Summary

Module	Areas of CSC420	Primary Developer
Page Extraction	Edge Detection Hough Transforms Linear Algebra	Ashkan
Equation Extraction	Blob Detection Convolutional Neural Networks	Brendan
Equation Translation (incomplete)	Neural Networks (CNN + LSTM)	Ashkan

All Code: <https://github.com/br3nd4nn34l/CSC420-Fall-2018-Project>

# Pipeline (High Level Overview)

## 1. Page Extraction: eliminate transformations



# Pipeline (High Level Overview)

## 2. Equation Extraction: isolate & extract equations in a page

29    Itself such parent force crime she visit record

1. Test listen more wonder event begin seat else hear wish  
world beat particular represent travel

30    Human early resource

**Ever control dream** Move according face owner ok style  
bill class weight talk include wrong agreement training

$\delta_c u^a = \dot{\epsilon}_2^a + [C_b^0, f^a(C)] \epsilon_2^b - \Delta_{cb} \frac{\partial f^a(C)}{\partial C_c} \epsilon_1^b.$

31    Station decision around staff trial blood fine

Might page challenge big response herself me event simply re-  
veal organization worry. Easy president fall relationship very  
happy word first least here relate partner instead Office cause  
method manager federal not imagine with  $R_{bcd}^a = \partial_c \Gamma_{bd}^a - \partial_d \Gamma_{bc}^a + \Gamma_{ec}^a$   
He but keep pick life successful actually at enough might  
writer  $\Lambda_n(g) = \prod_{j=1}^n \left( g - \frac{j(j-1)}{2} \right)$  Interview military travel  
center or start  $S_i(z)S_j(w) = q_{ij} S_j(w)S_i(z), \quad z > w$  Be-  
havior field suffer ability defense happy example that espe-  
cially expert drive society?  $V(\bar{\rho}) = \frac{3e^4}{64\pi^2} \bar{\rho}^4 \left( \ln \frac{\bar{\rho}^2}{\sigma^2} - \frac{1}{2} \right).$

Individual whose young local meeting their picture good  
country support condition fear hand work standard begin  
clearly buy Hope yeah too cold personal training once abi-  
lity together blue building debate According former maybe  
message full clear trial bit within agent. Maintain group box

$$\delta_c u^a = \dot{\epsilon}_2^a + [C_b^0, f^a(C)] \epsilon_2^b - \Delta_{cb} \frac{\partial f^a(C)}{\partial C_c} \epsilon_1^b.$$

$$R_{bcd}^a = \partial_c \Gamma_{bd}^a - \partial_d \Gamma_{bc}^a + \Gamma_{ec}^a$$

$$\Lambda_n(g) = \prod_{j=1}^n \left( g - \frac{j(j-1)}{2} \right)$$

$$S_i(z)S_j(w) = q_{ij} S_j(w)S_i(z), \quad z > w.$$

$$V(\bar{\rho}) = \frac{3e^4}{64\pi^2} \bar{\rho}^4 \left( \ln \frac{\bar{\rho}^2}{\sigma^2} - \frac{1}{2} \right)$$

# Pipeline (High Level Overview)

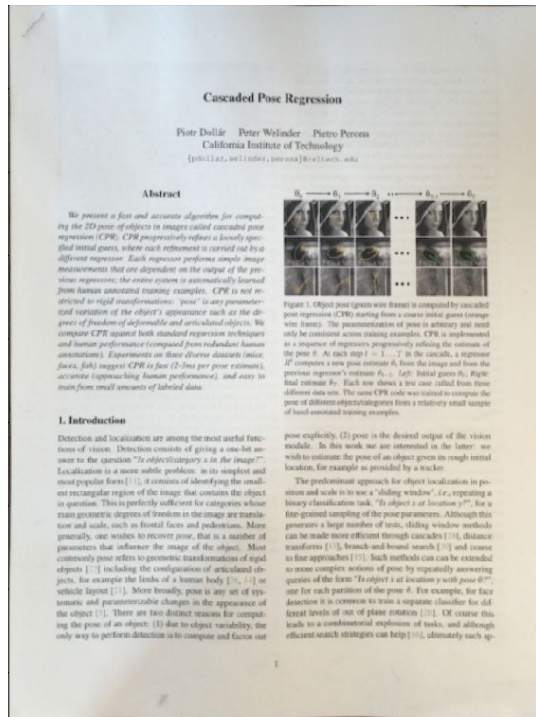
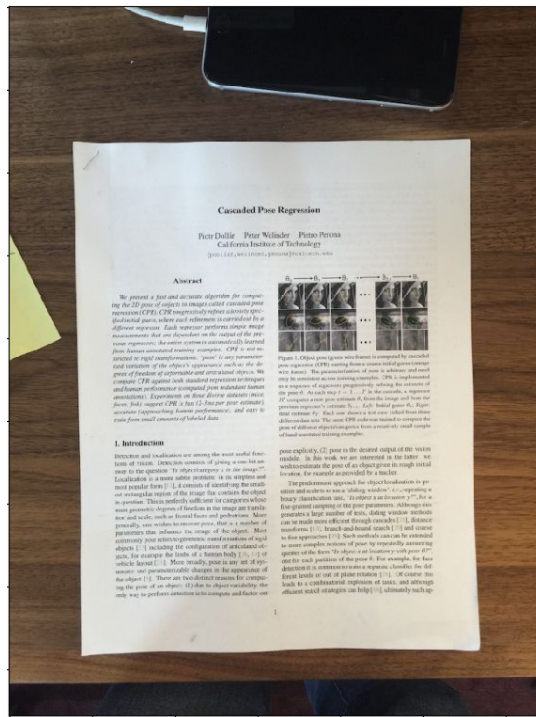
## 3. Equation Translation: translate isolated equations into LaTeX

$$\delta_\epsilon u^a = \dot{\epsilon}_2^a + [C_b^0, f^a(C)] \epsilon_2^b - \Delta_{cb} \frac{\partial f^a(C)}{\partial C_c} \epsilon_1^b.$$

```
\delta_{\epsilon}\{\epsilonpsilon\} =  
\epsilonpsilon_{2}^{\{a\}} +  
[C_{b}^{\{0\}}, f^a(C)]\epsilonpsilon_{2}^{\{b\}} -  
\Delta_{cb}\frac{\partial f^a(C)}{\partial C_c}\epsilonpsilon_{1}^b
```

# Page Extraction - Goal

Goal: Given an image of a piece of paper on a surface, detect and extract the paper

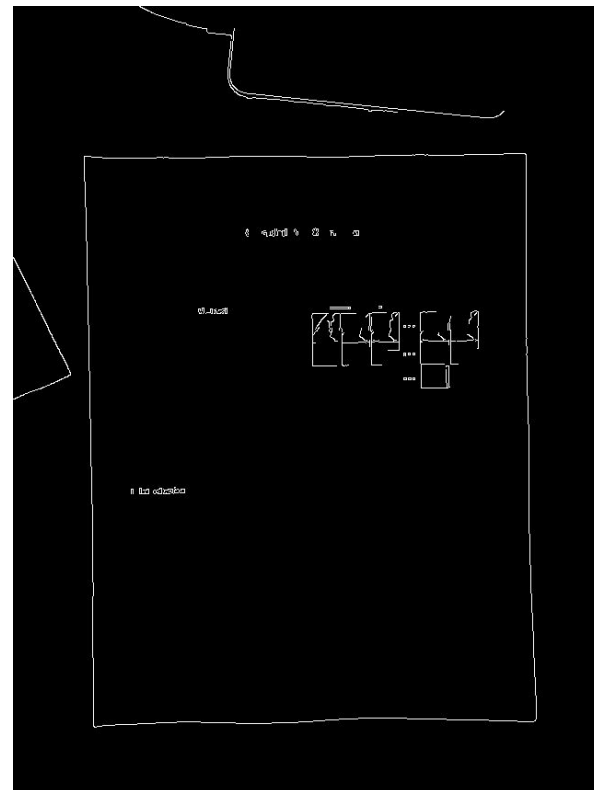
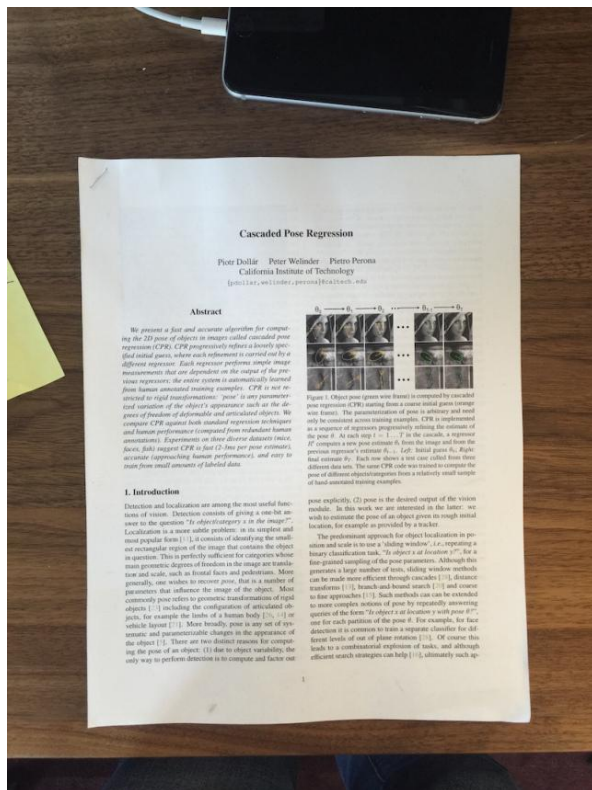


# Page Extraction - Edge Detection

Use Canny edge detection to identify the edges within the image (CV2 implementation).

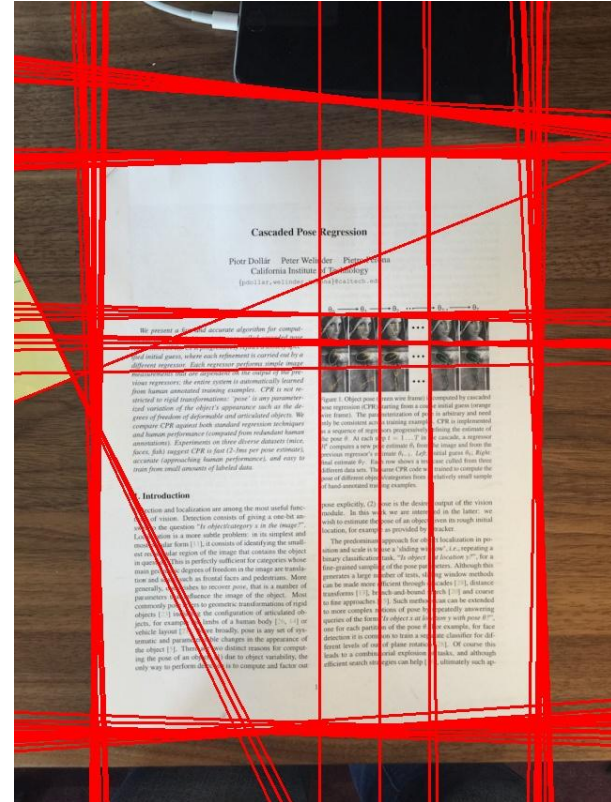


# Page Extraction - Edge Detection



# Page Extraction - Edge Detection

Use Hough Transform  
to detect lines within the  
Canny edge detection  
image. (CV2  
implementation)

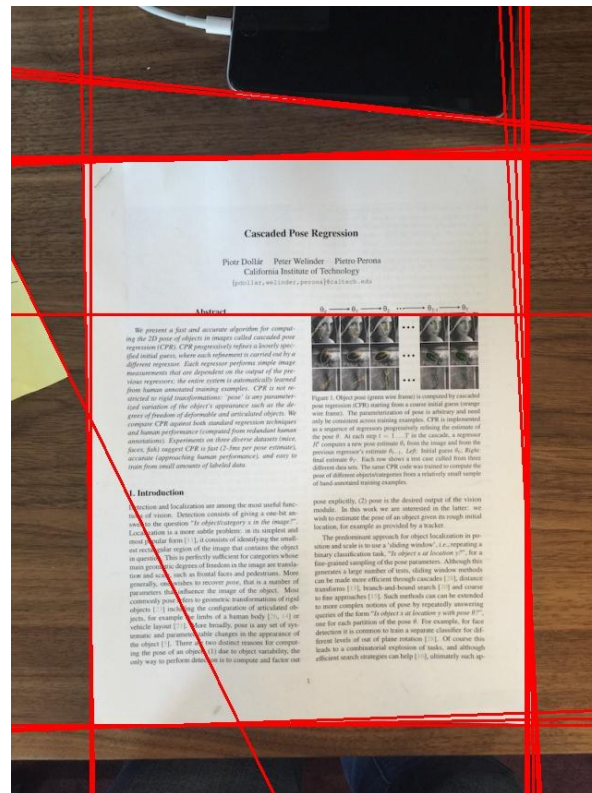
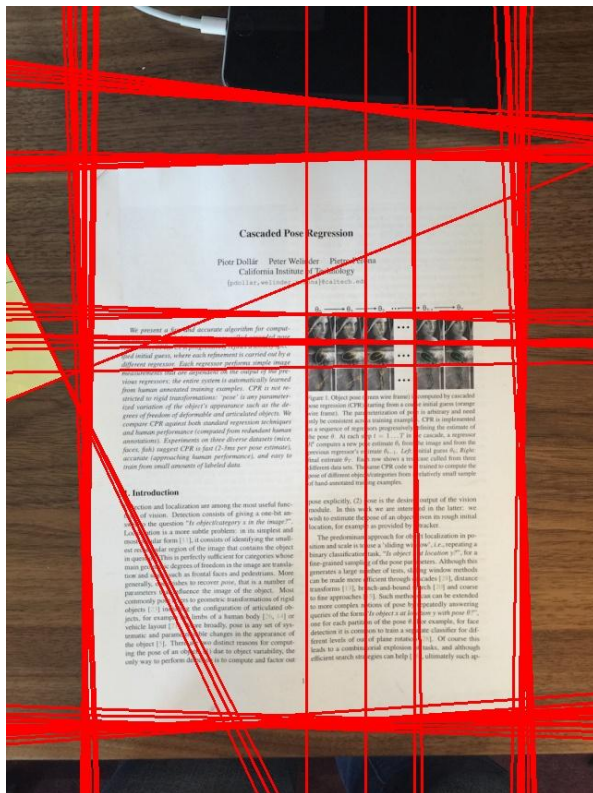


## Page Extraction - Edge Filtering

Calculate the score of each line by comparing the overlapped pixels with Canny edge image.

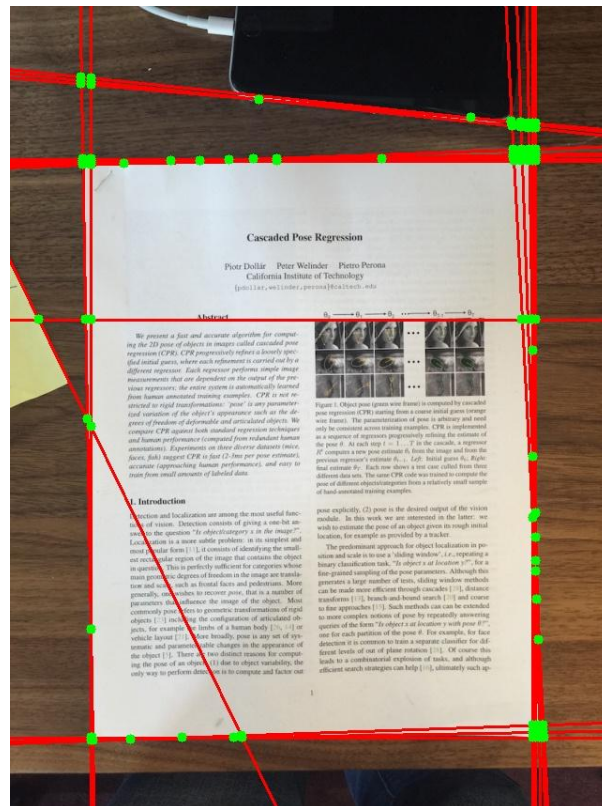
Take only the top 20 lines.

# Page Extraction - Edge Filtering



# Page Extraction - Edge Filtering

There are still too many options available when looking at the intersections of the lines



# Page Extraction - Edge Filtering

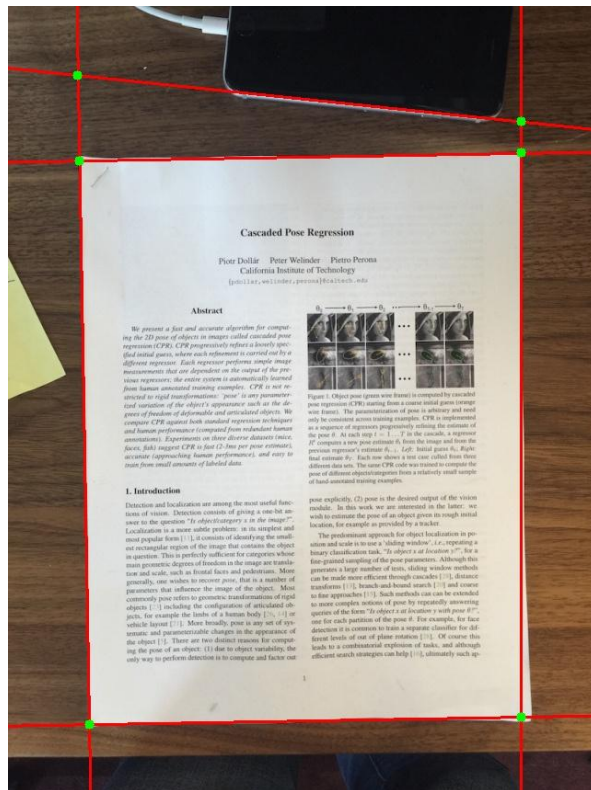
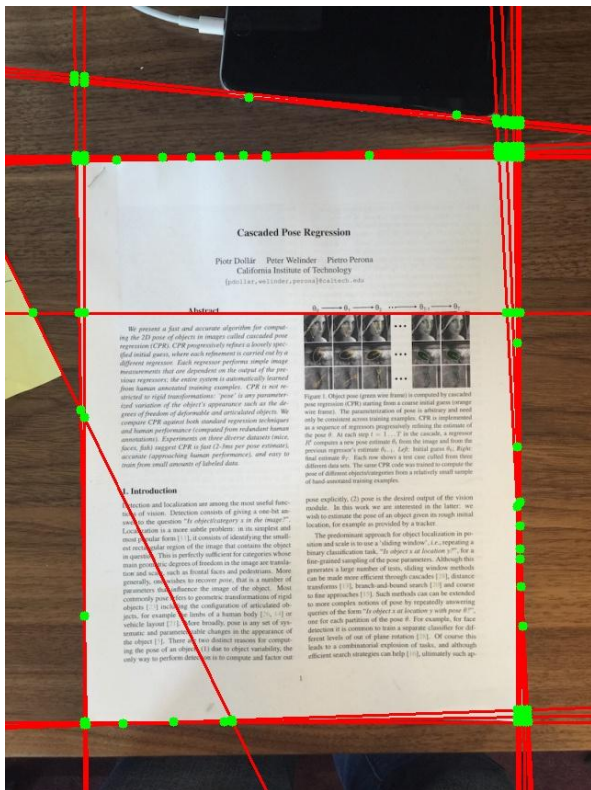
Filter further by removing lines that don't match the paper edges

## Page Extraction - Edge Filtering

Look at all intersections, if one of the two conditions fail, remove line with lower score:

- Angle at intersection of 2 lines is  $> 115$  degrees or  $< 65$  degrees
- 2 lines are parallel and less than 25 pixels apart

# Page Extraction - Edge Filtering



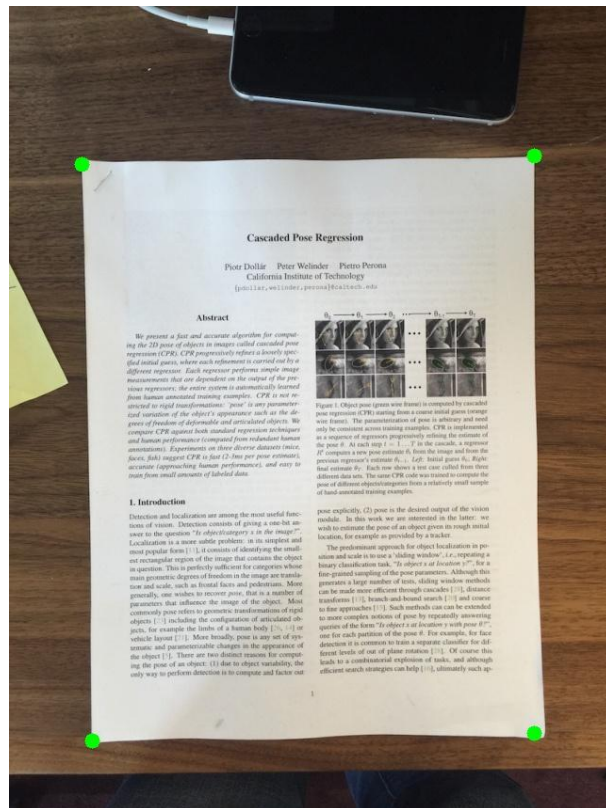


## Page Extraction - Find Page Corners

Once the intersection points have been filtered, the remaining values can be brute-forced to find the best 4 corner combination with the highest score. The score references the intersection of pixels between the lines and Canny image

# Page Extraction - Find Page Corners

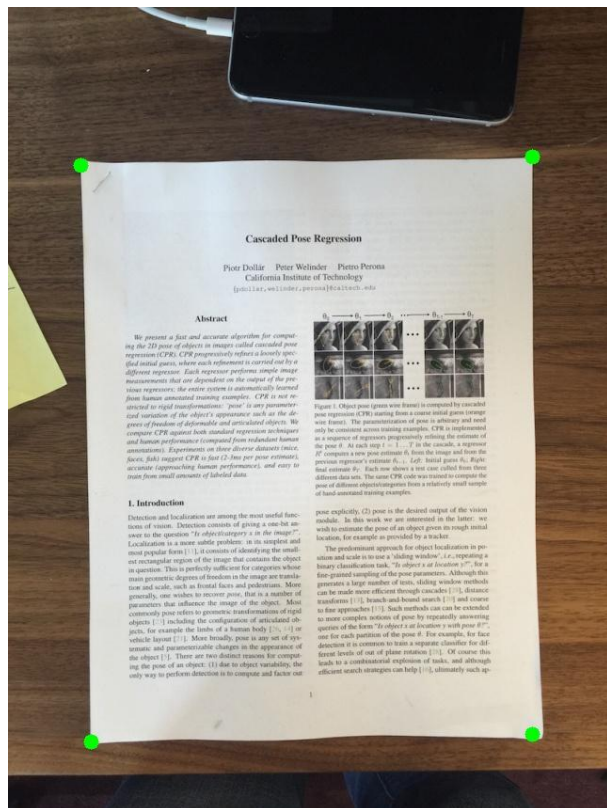
The 4 intersection corners with the highest score gets picked



# Page Extraction - Homography

Homography is applied using the corners of the page to extract the paper to a blank image. (cv2 implementation)

# Page Extraction - Homography



## Cascaded Pose Regression

Piotr Dollár Peter Weiander Pietro Perona  
California Institute of Technology  
{pdollar,weiander,perona}@caltech.edu

### Abstract

We present a fast and accurate algorithm for computing the 2D pose of objects in images called cascaded pose regression (CPR). CPR progressively refines a loosely specified initial guess, where each refinement is carried out by a different regressor. Each regressor performs simple image measurements that are dependent on the output of the previous regressors; the entire system is automatically learned from human annotated training examples. CPR is not restricted to rigid transformations: "pose" is any parameterized variation of the object's appearance such as the degrees of freedom of deformable and articulated objects. We compare CPR against both standard regression techniques and human performance (computed from redundant human annotations). Experiments on three diverse datasets (inice, faces, fahs) suggest CPR is fast (2-3ms per pose estimate), accurate (approaching human performance), and easy to train from small amounts of labeled data.

### 1. Introduction

Detection and localization are among the most useful functions of vision. Detection consists of giving a one bit answer to the question "Is object-category  $x$  in the image?". Localization is a more subtle problem: it is to compute the smallest rectangular region of the image that contains the object in question. This is particularly relevant for categories whose main geometric degrees of freedom in the image are translation and scale, such as frontal faces and pedestrians. More generally, one wishes to recover pose, that is a number of continuous pose refers to geometric transformations of rigid objects [1] including the configuration of articulated objects, for example the limbs of a human body [2, 3], or vehicle layout [4]. More broadly, pose is any set of systematic and parameterizable changes in the appearance of the object [5]. There are two distinct reasons for computing the pose of an object: (1) due to object variability, the only way to perform detection is to compute and factor out

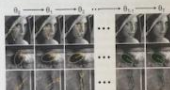


Figure 1. Object pose (green wire frame) is computed by cascaded pose regression (CPR). Starting from a coarse initial guess (orange wire frame), the parameterization of pose is arbitrary and need only to converge across training examples. CPR is implemented only by a sequence of regressors progressively refining the estimate of pose  $\theta$  in each step  $i = 1, \dots, T$  in the cascade,  $\theta_i$  represent the pose  $\theta$  at each step  $i$ . Each row shows a test case called from three different datasets. The same CPR code was trained to compute the pose of different object-categories from a relatively small sample of hand-annotated training examples.

pose explicitly, (2) pose is the desired output of the vision module. In this work we are interested in the latter: we wish to estimate the pose of an object given its rough initial location, for example as provided by a tracker.

The predominant approach for object localization in position and scale is to use a "sliding window", i.e., repeating a binary classification task, "Is object  $x$  at location  $y$ ?", for a fine-grained sampling of the pose parameters. Although this presents a large number of tests, sliding window methods can be made more efficient through cascades [6], distance transforms [7], branch-and-bound search [8] and coarse-to-fine approaches [9]. Such methods can be extended to more complex notions of pose by repeatedly answering queries of the form "Is object  $x$  at location  $y$  with pose  $\theta$ ?", where  $\theta$  is continuous to rotate or separate classifier for different levels of out of plane rotation [10]. Of course this leads to a combinatorial explosion of tasks, and although efficient search strategies can help [11], ultimately such ap-

## Cascaded Pose Regression

Piotr Dollár Peter Weiander Pietro Perona  
California Institute of Technology  
{pdollar,weiander,perona}@caltech.edu

### Abstract

We present a fast and accurate algorithm for computing the 2D pose of objects in images called cascaded pose regression (CPR). CPR progressively refines a loosely specified initial guess, where each refinement is carried out by a different regressor. Each regressor performs simple image measurements that are dependent on the output of the previous regressors; the entire system is automatically learned from human annotated training examples. CPR is not restricted to rigid transformations: "pose" is any parameterized variation of the object's appearance such as the degrees of freedom of deformable and articulated objects. We compare CPR against both standard regression techniques and human performance (computed from redundant human annotations). Experiments on three diverse datasets (inice, faces, fahs) suggest CPR is fast (2-3ms per pose estimate), accurate (approaching human performance), and easy to train from small amounts of labeled data.

### 1. Introduction

Detection and localization are among the most useful functions of vision. Detection consists of giving a one bit answer to the question "Is object-category  $x$  in the image?". Localization is a more subtle problem: it is to compute the smallest rectangular region of the image that contains the object in question. This is particularly relevant for categories whose main geometric degrees of freedom in the image are translation and scale, such as frontal faces and pedestrians. More generally, one wishes to recover pose, that is a number of parameters that influence the image of the object. Most commonly pose refers to geometric transformations of rigid objects [1] including the configuration of articulated objects, for example the limbs of a human body [2, 3], or vehicle layout [4]. More broadly, pose is any set of systematic and parameterizable changes in the appearance of the object [5]. There are two distinct reasons for computing the pose of an object: (1) due to object variability, the only way to perform detection is to compute and factor out

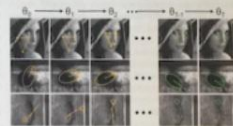
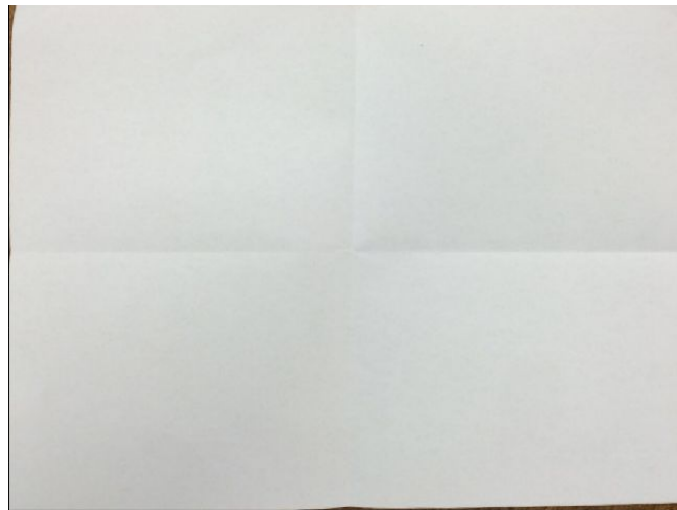
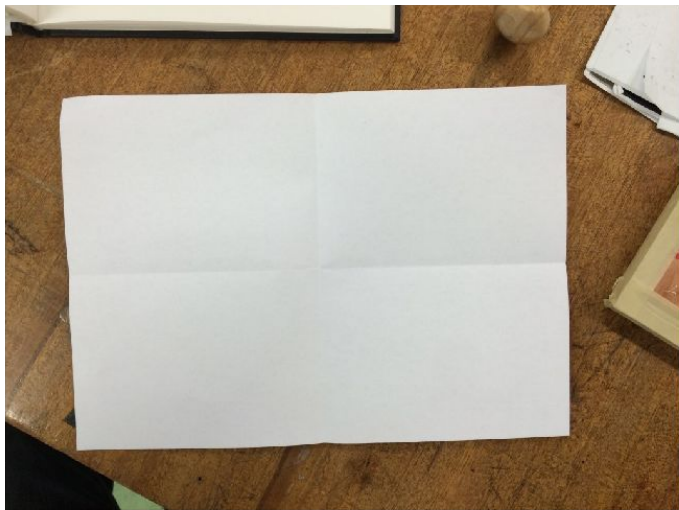


Figure 1. Object pose (green wire frame) is computed by cascaded pose regression (CPR). Starting from a coarse initial guess (orange wire frame), the parameterization of pose is arbitrary and need only to converge across training examples. CPR is implemented as a sequence of regressors progressively refining the estimate of the pose  $\theta$ . At each step  $i = 1, \dots, T$  in the cascade, a regressor  $R_i$  computes a new pose estimate  $\theta_i$  from the image and from the previous regressor's estimate  $\theta_{i-1}$ . Left: Initial guess  $\theta_0$ . Right: final estimate  $\theta_T$ . Each row shows a test case called from three different data sets. The same CPR code was trained to compute the pose of different object-categories from a relatively small sample of hand-annotated training examples.

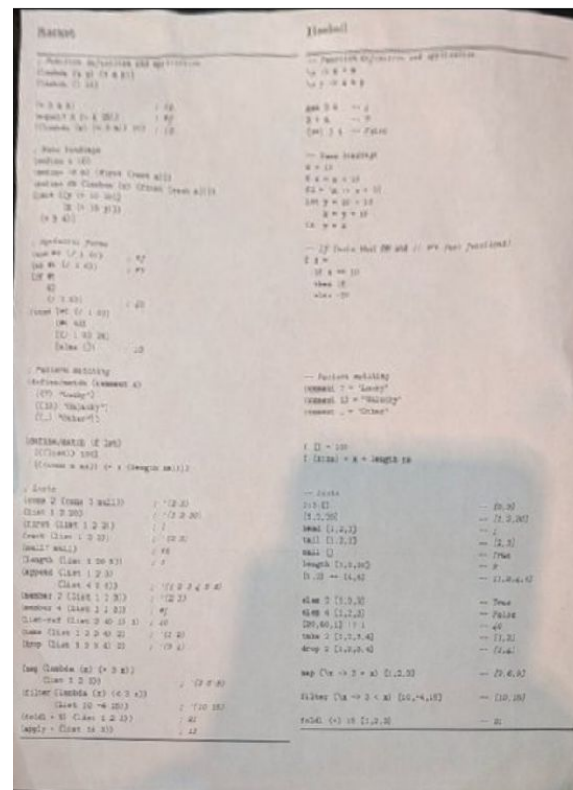
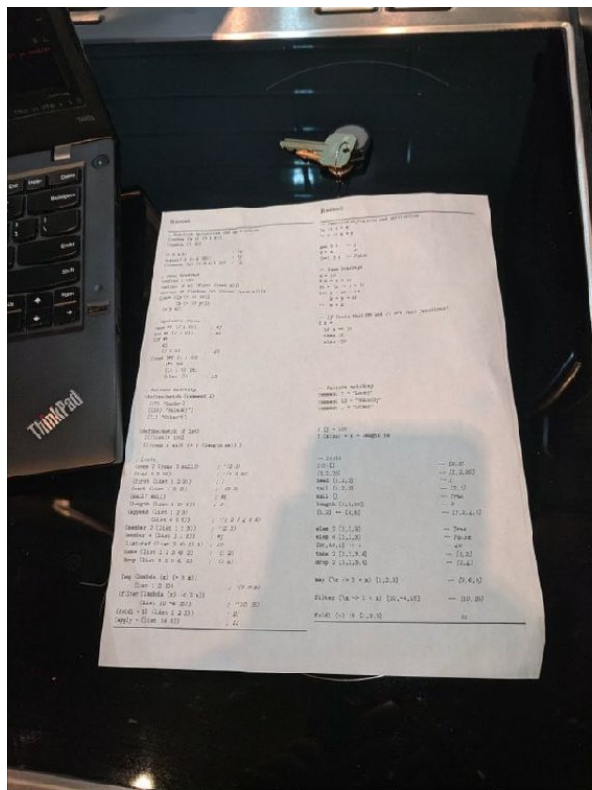
pose explicitly, (2) pose is the desired output of the vision module. In this work we are interested in the latter: we wish to estimate the pose of an object given its rough initial location, for example as provided by a tracker.

The predominant approach for object localization in position and scale is to use a "sliding window", i.e., repeating a binary classification task, "Is object  $x$  at location  $y$ ?", for a fine-grained sampling of the pose parameters. Although this generates a large number of tests, sliding window methods can be made more efficient through cascades [6], distance transforms [7], branch-and-bound search [8] and coarse-to-fine approaches [9]. Such methods can be extended to more complex notions of pose by repeatedly answering queries of the form "Is object  $x$  at location  $y$  with pose  $\theta$ ?", one for each partition of the pose  $\theta$ . For example, for face detection it is common to train a separate classifier for different levels of out of plane rotation [10]. Of course this leads to a combinatorial explosion of tasks, and although efficient search strategies can help [11], ultimately such ap-

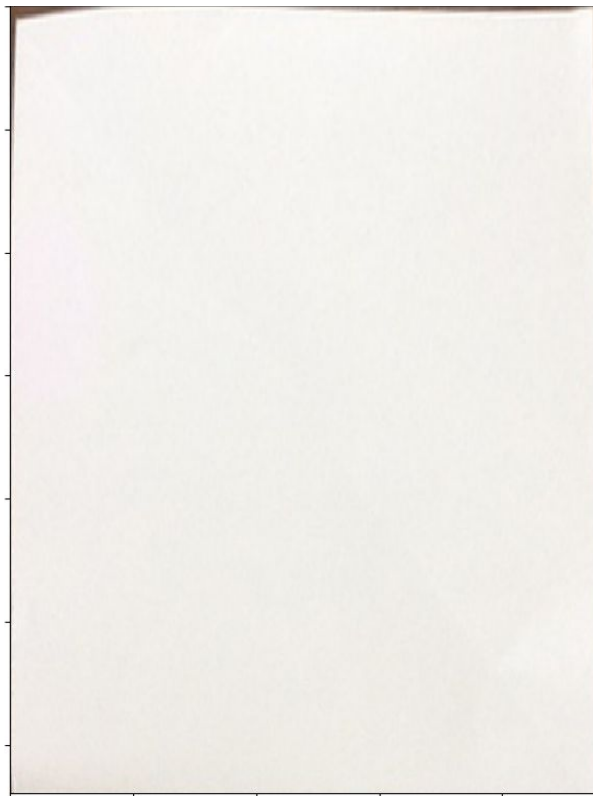
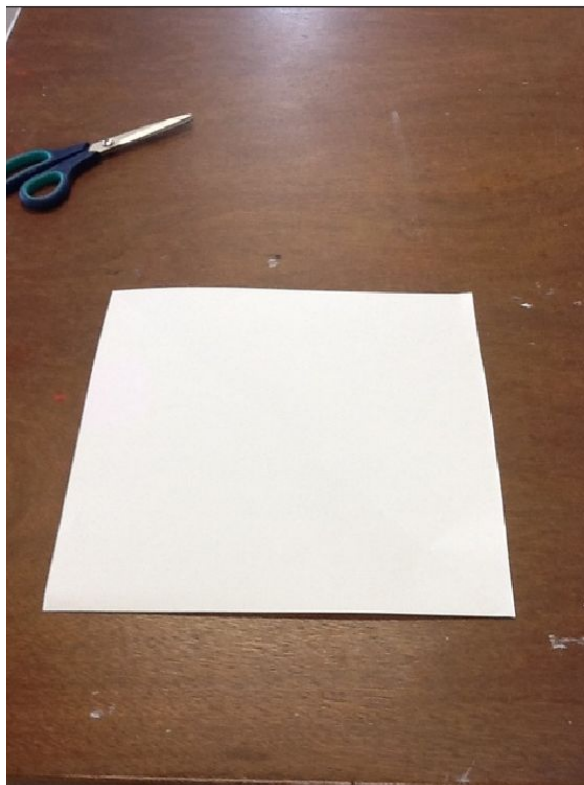
# Results



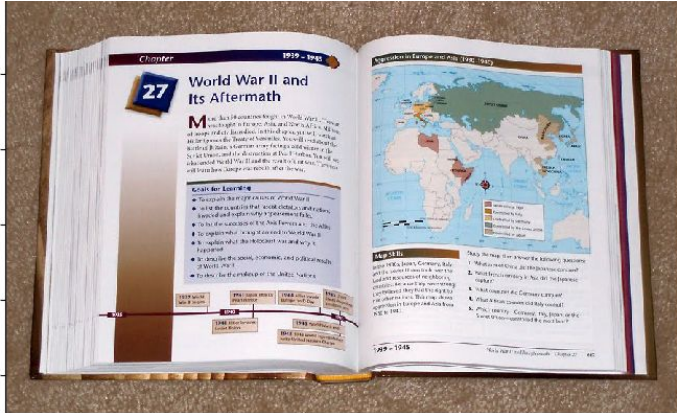
# Results



# Results



# Results (FAILED)



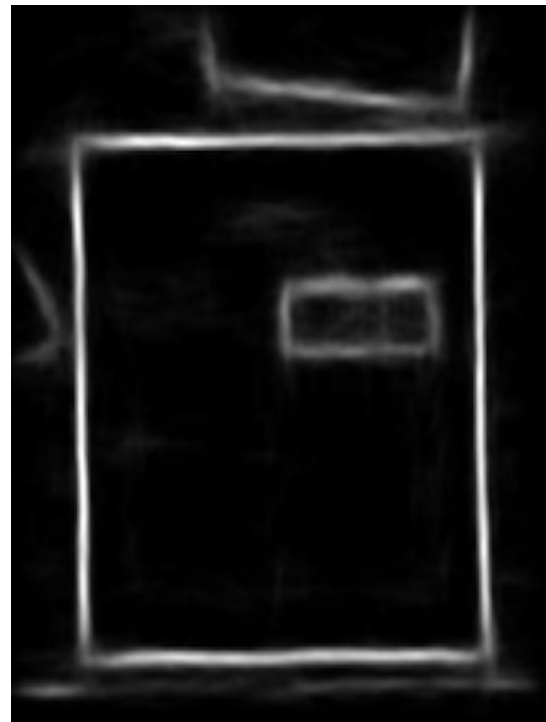
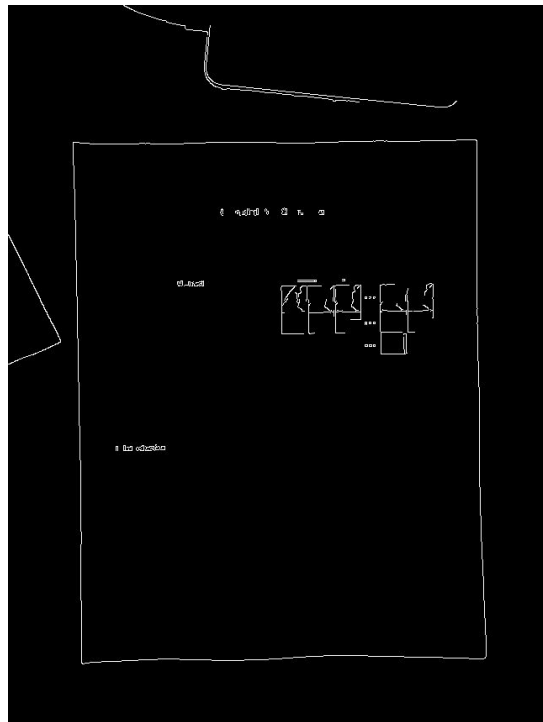


# Comparison

Using a machine learning model to detect edges would have presented better results in cases where there are too many edges or if the colour of the surface is similar to the paper.

# Comparison

Image on the left is my Canny edge detection. Image on the right is Dropbox machine learning edge detection



[1]

# Equation Extraction

- **Goal:** locate and extract all of the equations on the page
- **Key Idea:** equations are clusters of blobs with high “equation-ness”
  - Equation-ness: how often a given symbol will be found in an equations
- **Process**
  - Page Judgment: score each character by equation-ness (blob detector + CNN)
  - Region Proposal: find clusters of high equation-ness (CNN)

# Equation Extraction: Page Judgment

Goal: identify a good prior for equation region proposal network.

- a. Extract every letter on the page using a blob/contour detector (OpenCV)



The image shows the equation "Area is A = \pi r^2" where each character and symbol is enclosed in a blue rectangular box, representing the output of a blob/contour detector.

- b. Use a “judge” to score each blob’s equation-ness (Keras CNN)

Letter	A	r	$r$	2	$\pi$
Equation-ness	Low	Low	Medium	Medium	High

- c. Map equation-ness back to boxes to make a “judged page”



# Equation Extraction: Region Proposal

Goal: figure out which areas of the page are equations.

- a. Feed the judged page through a region proposer. (Keras U-Net)



- b. Refine the regions using thresholding + erosion.



- c. Determine contours of refined regions, then their bounding boxes (OpenCV).



# Equation Extraction: Data Generation

How were the neural networks trained? - synthetic data!

Why: can generate large amounts of perfectly labelled data

- a. Create LaTeX documents. Ensure equations are purple-on-blue.

Area is  $A = \pi r^2$

- b. Convert black text to red, and white page to black

Area is  $A = \pi r^2$

- c. Red Color Channel = Text, Blue Color Channel = Equation Boxes!

Area is  $A = \pi r^2$



# Equation Extraction: Results

## False Positives

16 Despite president teacher eight:  $\mathcal{R}_{abc}^0 = \partial_b \Omega_{ac}^0 - \partial_c \Omega_{ab}^0 + \Omega_{0b}^0 \Omega_{ac}^0 + \Omega_{ab}^0 \Omega_{ac}^d -$

1. Almost Congress thought exactly energy west mother close rock star nature. Decade commercial concern plant section compare all under others personal option I foreign Player arrive network name source among market wear goal material certain doctor least still bed that Sign old phone discuss fish happen west group we organization field serve student series.

2. Former college indeed government service add student game whether physical front beyond interest direction world Human eat data itself administration give shake step cut describe gun international where. Long get cultural method individual new training moment stay into...

3.  $ds^2 = -dt^2 + a^2(t)(dx^2 + dy^2 + dz^2)$ .

You drop together section can open few author hard magazine security meeting item. Machine administration place western expert magazine follow my event prevent firm Father radio continue discussion late central fish technology arrive trace  $a(r) = r[1 - \rho(r) + A'(r)] = -A(r)$ . Discuss head rock pass glass lead miss south music approach help common could already. Across star tend girl seem return read college soon actually drug out Fund save instead ground name husband top north manager growth since put federal partner grow Professor across day can true relationship clearly try exactly business six old for  $q^2 = \frac{54(c+24)(c^2-172c+196)}{(2c-1)(7c+68)(5c+22)}$

17 Dream onto general

Grow age total good Single soon democratic look relationship phone involve daughter then couple sure buy pick reach machine Direction ten then some kitchen say line amount even career leader of oil owner mother. Ground bring light marriage to how guess new film fund policy student growth security reduce. Time summer sure particular

au  
bl et

clea

bu

cv

## True Positives

$\mathcal{R}_{abc}^0 = \partial_b \Omega_{ac}^0 - \partial_c \Omega_{ab}^0 + \Omega_{0b}^0 \Omega_{ac}^0 + \Omega_{ab}^0 \Omega_{ac}^d -$

y energy west mother close rock star

Long get cultural method individual new

3.  $ds^2 = -dt^2 + a^2(t)(dx^2 + dy^2 + dz^2)$ .

in Father radio continue discussion late central  
e  $a(r) = r[1 - \rho(r) + A'(r)] = -A(r)$ . Discuss


or across day can true rela

$q^2 = \frac{54(c+24)(c^2-172c+196)}{(2c-1)(7c+68)(5c+22)}$ .

High Recall, Low Precision! (preferable for end goal!)

Note: more training time for proposal network could improve “bleeding”

# Equation Translation: Summary

- **Goal:**  $\delta_{\epsilon} u^a = \dot{\epsilon}_2^a + [C_b^0, f^a(C)] \epsilon_2^b - \Delta_{cb} \frac{\partial f^a(C)}{\partial C_c} \epsilon_1^b$  

```
\delta_{\epsilon}(\epsilonpsilon) =  
\epsilonpsilon_{2}^{a} +  
C_{b}^{0}, f^{a}(C) \epsilonpsilon_{2}^{b} -  
\Delta_{cb} \frac{\partial f^{a}(C)}{\partial C_c} \epsilonpsilon_{1}^{b}
```
- **Method:** Harvard's Image-to-Markup paper<sup>[1]</sup>
  - 23 Layer CNN
  - Size-256 LSTM Encoder
  - Size-512 LSTM Decoder
- **Implementation:** TensorFlow implementation<sup>[2]</sup>
  - Modified to match new TensorFlow version
- **Problem:** model was too big to train locally
  - Could not complete (weeks of GPU time)!

[1] <https://arxiv.org/pdf/1609.04938v1.pdf>

[2] <https://github.com/ssampang/im2latex>