Topic 2: Brainstorming

Possible Approaches:
Start with Object Detection: Recognize multiple objects within the image (common methods when given an image, it generates predicted object with cio?, the others generate binary output (either present or not), we could use 'ReLu' as the corresponding probability). In our topic, only the 'violence' category requires a probability activation function for the second to last layer to achieve this.

Need to solve:

- Find available library that we can adapt to our model (preferably ones that are open sourced)
- Data preprocessing: format the given input dataset so they can be adapted to the library functions
- Make performance adjustments, model tuning
- Think of/ implement ways to make improvements from base model
- Format output

Difficulty:

- Object detection methods can only efficiently label objects such as person, hat, traffic light
- Give the ground truth within both training set(annot_train.txt) and test test(annot_test.txt), we should determine whether our analysis is meaningful.
- It's difficult to detect contextual information, training such as 'violence ratio', 'protest' that cannot be defined by a single object (but set's prelabrour own object list alongwith 0s/1s. Combine it with training set to train our first neural net.

Step2: Based on the output of our first NN, we train our second neural network and output violence rate.

Reference(Code / Articles / Paper):

1. Projects made using Tensorflow: https://github.com/jtoy/awesome-tensorflow
2. https://github.com/rbgirshick/fast-rcnn/tree/master/data
3. Repo for Papers with code: https://github.com/zziz/pwc
4. Paper on Selective Search: https://ivi.fnwi.uva.nl/isis/publications/2013/UijlingsIJCV2013/UijlingsIJCV2013.pdf
5. https://cs.stanford.edu/people/karpathy/rcnn/
6. Faster-RCNN essay: https://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf
   Demo:https://towardsdatascience.com/object-detection-with-10-lines-of-code-d6cb4d86f606

   7."Protest Activity Detection and Perceived Violence Estimation from Social Media Images":

Topic 2: Brainstorming

Understanding protest and violence has been an overwhelmingly critical topic within research in political science, there are great efforts attempted to tackle these “social multimedia and computer vision” problems: Liu[1] applied facial attribute classification to understanding appeals from protester; You and other researchers[2] tried to analyze photographs and portrayals of particular politicians; Public opinions and presentations in social media are also considered in processing emotion, mobilization as well as immigration[3]. Thanks for these work to progressing human perception of media content as well as taking more advantages to recognize subjectives.

We choose UCLA_PROTEST_IMAGE_DATASET as training set since the annotation for UCLA dataset is fulfilled and has extra labels such as Group_20 and Group_100 that helps detect emotional sentiment thus improving prediction accuracy for violence and protest estimation. The UCLA dataset contains 40,764 images, among which 11,659 images are protest images.

Here is how we define the annotation:
Protest: Whether there is a current protest in the image or not.
Violence: How violent the image shows. It is more an estimation and evaluation than exact numbers.
Sign: Whether a protester is holding a visual sign or not(maybe on paper, panel or wood).
Photo: Whether a protester is holding a photograph or an individual or not.
Fire: Whether there is smoke or fire in the scene.
Police: Whether there are policemen or troop in the scene.
Children: Whether there are children involved in the scene.
Flag: Whether there are flags in the scene.
Night: Whether it is during the night-time or in the daytime.
Shouting: Whether there are people shouting in the scene.

According to Won’s paper[4], we choose our model based on a 50-layer ResNet, which consists of 50 convolutional layers with batch normalization and ReLU layers. The features computed through convolutional layers are shared by linear layers for multiple classification(i.e., protest, violence, sign, photo, fire). After setting training epoch up to 20, we can get prediction accuracy at 94%.

[1] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. 2015. Deep learning face attributes in the wild. In Proceedings of the IEEE International Conference on Computer Vision. 3730–3738.
[2] Quanzeng You, Liangliang Cao, Yang Cong, Xianchao Zhang, and Jiebo Luo. 2015. A multifaceted approach to social multimedia-based prediction of elections. IEEE

Transactions on Multimedia 17, 12 (2015), 2271–2280.
[3] Lefteris Anastasopoulos and Jake Williams. 2016. Identifying violent protest activity with scalable machine learning *. (2016). http://scholar.harvard.edu/janastas
[4] Donghyeon Won, Zachary C. Steinert-Threlkeld, Jungseock Joo. 2017. Protest Activity Detection and Perceived Violence Estimation from Social Media Images. In Proceedings of the 25th ACM International Conference on Multimedia 2017.

(temp) Maya’s take on topic proposal:
Overall dataset contains lots of protesting events, political figures, then random pictures with texts, can focus on **event analysis** (**what kind, violence, emergency level** (should public be alerted if such image present on social media somewhere, should police force be requested; given nowadays journalists are not the first ones to record an emergency event but any civilians with phones and they send to social media almost immediately for sharing). Facebook also has alert function, ex: friends marked safe in x incidents in y place, how Facebook get notified of such events? Or shorten police response time (without needing a call, could event detection systems detect such event happening in proximity and alter police force) etc.

Extended topics:

- If an event’s taking place, what kind of event? Protest, Sport, meeting, music, group taking photo together. (need extra labeling effort)

- (preferred) Whether a protest is taking place, **give violence and emergency level** : in addition to violence ratio, also include presence of Children, Fire, shouting, set Children, fire presence to high emergency, shouting to medium. (for Children, to avoid misinterpreting a happy event as a disaster, first need to determine that the pic contains groups of people and violence ratio is not low). Can also include ‘police’ (extension likely not gonna do for this proj.: if police are presented, do they need backup given level of violence of event?).

‘flag’ and ‘night’ are also good but can be misleading sometimes, ex: sports events have flags, romantic or friends celebrations can happen at night. ‘Sign’ is a good one for protest too, but a road sign is also labelled as a sign and has nothing to do with protests.

Maybe we could use how to balance and use these ambiguous features in identifying a protesting event to fulfill the requirement ‘improvement from baseline solution’. But this is hard and we’re not experts in this field, however, the signs/features used (children, fire, violence) are pretty common sense, we can go from the angle ‘as civilians, when facing these situations, do we feel alert?’, it’s a valid perspective/justification.

Other possibilities:

Topic 2: Brainstorming

- Categorizing photo based on image content (put photos into categories): specific for Trump, ISIS, specific for a certain human right (ex: LGBT, equal right, abortion etc.) (need extra labeling effort)

- capturing sentences/words on signs, interpret situation/ context based on text retrieved. (eg: funny poster, propaganda, protesting event) (hard, likely not finish in 2 weeks)

Reference(Code / Articles / Paper):
A basic nearest neighbor classifier: http://cs231n.github.io/classification/

Docs:

1)
Bounding box labeling versus Tag label classification

2)
Without bounding box, can we train a single neural net that can classify several objects and output their presence?

3)
A neural network combined with an expert system.

First, We train a neural network. Inputs: picture and its corresponding label. Ideally we can identify some useful labels we need, such as, fire/night/child/violence.
Our goal is to use our classifier to extract similar labels from pictures in the test set.

Secondly, we build an expert system, which has several emergency levels. The input is the output of our neural network, while the output matches the emergency levels. With this expert system, it's easier for police to decide if they need to help preserve public order and maintain a safe environment.

4) OCR: Do we need to use a bounding box to train the net? Are there any available library/framework we can adapt.

Proposed Approach using pre-trained Neural Network for Object Detection
Brief Problem Definition:

- Why we're choosing this: nowadays journalists are not the first ones to record an emergency event but any civilians with phones, and they send to social media almost immediately when such event takes place in their proximity

Topic 2: Brainstorming

- What's our goal: We'd like to build a system that looks for such event in a pool of images (could be social media feeds, we do not extend to data mining from such source in this project, but purely training a NN and an expert system from a given image pool), and output either a ratio or a binary conclusion for 'emergency level' and whether law enforcement should be requested.
- Justification: we do not wish to build a system that can define whether police should react or not, since we're not expert in criminal activities etc., but from a civilian's perspective, if such things happen around us, do we feel fear or panick, and do we wish law enforcement would respond in a timely manner.

Step 1: Feature Selection (ones that are detectable by NN)

- Protest (hard to define, can either use 'Signs' or 'by #of people detected' or 'flag')
    - If using 'Signs' to represent 'Protest' we can first do a correlation check between the two based on provided labels
    - Detecting # of people could be a complex task
    - If using 'Flag' can also do correlation check
    - Proposed solution:
        - Check correlation between 'sign' and 'protest', as well as 'flag' and 'protest', use the one yield higher correlation to represent 'Protest'
- Fire
- Other possibilities: Police, Children, but we can start from the above two as baseline approach and go from there.

Step 2: Data Preprocessing

- Image resize, most frameworks require specific input size and format (shouldn't be difficult to do)
- Reduce sample size if don't want to run on GPUs, but need to ensure we have a good balance of positive and negative examples, possible approaches:
    - (must) Eliminate ones that do not have labels for the features we choose.
    - Random sampling or
    - Choose based on violence ratio, since we're interested in event that would create public panic, so we can select samples based on violence ratio (ex: positive if violence ratio >=0.5, negative if <= 0.1, neutral if in between, only guessing, need to do a Distribution plot for violence if want to use this)

Step 3: Search methods online, using pre-trained object recognition NN for our own input and output (it's possible to do), helpful links:

- https://towardsdatascience.com/keras-transfer-learning-for-beginners-6c9b8b7143e
- https://www.analyticsvidhya.com/blog/2018/07/top-10-pretrained-models-get-started-deep-learning-part-1-computer-vision/
- https://machinelearningmastery.com/use-pre-trained-vgg-model-classify-objects-photographs/

Topic 2: Brainstorming

- https://www.quora.com/What-is-the-VGG-neural-network (answer 1 is helpful, Betke mentioned VGG 16 layers as a possibility)
- http://www.robots.ox.ac.uk/~vgg/demo/
- Many more, terms I used for search:
    - Use pre-trained neural network with your own input and output
    - Transfer learning
    - Multi-object recognition in image using VGG/RestNet

Step 4: Try out one or more approaches, build baseline solution from there

- Most approaches suggest adding/modify our own density layers on top of model, change output layer, possible tool: Keras (has a method to load pre-trained NN, first link above)
- what Betke suggested after class: keep the 2nd to last layer and modify the last layer to yield our desired output

Step 5: Improve on baseline

- Training pre-existing NN models to take our own input and output should already be counted as part of the improvement
- Secondly, we add our own Expert System based on NN's output to generate the 'public panicking'/ 'emergency level' rating, can add different weight to diff detected feature, but we have only 2 here, so might not be needed
- Give reasoning why our system yield such result (build as part of the output, ex: print statement 'Panicking level: x, because a, b are detected'), given we only have 2 features, we might not be able to generate a level, so maybe go for 'public should be altered and avoid surrounding area' or not/ 'law enforcement should be altered' or not.

Other possible approaches, if Step 3 yield undesirable result:

- Text analysis from signs using VGG OCR (not that difficult if using pre-trained models actually, might even be easier than object recognition from images (our current approach): detect photos with signs, then do text analysis from there

- Facial expression analysis to detect fear, anger from protestants (probably not that hard either once we do nice preprocessing for input data)

Multi-Label Transfer Learning useful links:

w/Keras & VGG:
https://www.pyimagesearch.com/2018/05/07/multi-label-classification-with-keras/

w/Inception net:
https://towardsdatascience.com/multi-label-image-classification-with-inception-net-cbb2ee538e30

Topic 2: Brainstorming

Others:
https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/

https://www.tensorflow.org/hub/tutorials/image_retraining

https://medium.com/learning-machine-learning/multi-class-fish-classification-from-images-with-transfer-learning-using-keras-335125637544

https://cs231n.github.io/transfer-learning/

https://towardsdatascience.com/a-comprehensive-hands-on-guide-to-transfer-learning-with-real-world-applications-in-deep-learning-212bf3b2f27a

Topic 2: Brainstorming

March 26th

Some new ideas:

Step 1:
Use Bing image search API to build our own customized object training sets.

Step2:
Build a multi-label classification with Keras, based on our training sets.
Pros: It can identify combination of objects. For example, black dress / blue shirt. However, it cannot identify unknown combinations, like black shirt / blue dress.

Based on this, we considered another approach.
Given two objects, sign and fire. There are 2 ^ 2 = 4 combinations for their labels: 00, 01, 10 and 11. We can divide our training sets into these four categories, and train our neural network to accurately identify them. Furthermore, we can classify each picture into an interval of violence. For example, given a picture with both fire and signs, it can be classified into violence > 0.4

The problem we faced: Without bounding box labeled, if we take bing images as training sets, it's very likely that we cannot accurately localize the objects within our test set. We thought images returned by bing didn't have enough data needed to train the net.