### Objectives
- Develop an effective scoring algorithm to rank the most interesting webcams.
- Cluster webcams to find similar categories of webcams.
- Use a ConvNet to identify objects in webcam images to improve filtering relevance.

### Scoring Algorithm Approach
- **Uninteresting webcams**: Low/No activity
- **Interesting webcams**: Increasing/High activity

### Background Model
- Model background color of each pixel using a Gaussian Mixture Model (GMM)
- GMM needed to account for multi-modal background distributions, single Gaussian insufficient for some backgrounds.
- Online unsupervised learning – Background model adapts to changes in illumination / objects.
- Parameters updated recursively based on last T number of frames as training data, adaptive number of Gaussians.
- OpenCV BackgroundSubtractorMOG2 model based on background subtraction paper by Z. Zivkovic.
- Model classifies a pixel as background or foreground.

### Background Model
- Background Model returns foreground mask
- Fit contours around blobs in foreground mask, filtered by size to account for noise (overly small contours) and change in webcam global illumination / movement (overly large contours)

### Object Detection
- Score = (total contour area %)/(total contour length)
  - Max. area to show most/largest objects, Min. length to reward convexity (penalizes large noisy artifacts).
  - Sample displays of Top 5 / 1000 webcams shown below. Background model shown beneath webcam images with highlighted objects.

### Clustering Webcams
- Bag-of-features model used to represent images.
- Cluster SIFT descriptors from a subset of images to create visual vocabulary.
- Feature vector is a histogram of closest features in the visual vocabulary.
- K-means++ produced significantly better clusters than random initial clusters.
- Choice of K was observed to be important. Setting K too high resulted in noisy clusterings; setting K too low resulted in meaninglessly broad categories.

### Convolutional Networks for Filtering
- Start with a convolutional network pretrained on the ImageNet dataset.
- Modify output layer to classify humans (blue), vehicles (red), and noise.
- Create custom dataset by labeling patches found using object detection.
- Train pretrained model on our custom dataset.
- Use classifier to filter out noise during object detection.
- Applications for webcam search and clustering.