A Novel Approach for Predicting and Understanding Road Danger in the Developing World:
Deep Video-Classification of Roads in Nairobi, Kenya

Alexandr Lenk, Matias Cersosimo, Negin Raoof

Stanford University

OUR VIDEO

Click to watch the Poster video!

MOTIVATION

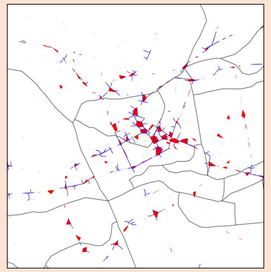
Providing traffic safety and lowering the rate of road accidents in Nairobi, Kenya is a major concern. With a road traffic death rate of 27.8 per 100,000 inhabitants in 2016, Kenya has nearly twice as much road fatalities as the world average.

Collaborating with the World Bank, we are the first ones to construct a deep learning model entirely based on videos of several road segments from Nairobi. The model allows us to analyze different road conditions and predict danger level of roads.

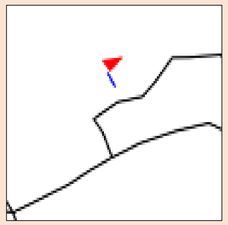
DATA PROCESSING AND MERGING

Data consists of:

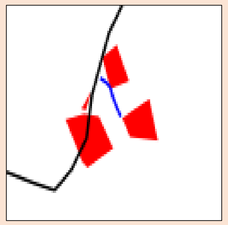
- Geospatial Dataset of 912 100-meter long road segments.
- Geospatial Dataset of 1428 crash hotspots linked to the number of annual crashes between 2012-2018.
- 852 road and pedestrian activity videos.



Center of Nairobi with roads (blue) and hotspots (red)



Road Segment matched to 1 Hotspot



Road Segment matched to 4 Hotspots

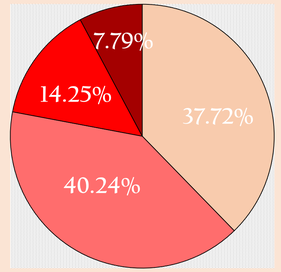
Following the World Bank data collecting strategy, we match road segment *I* to hotspot *j* as long as the distance between *i* and *j* is less than 130m.

LABELING STRATEGY AND CLASS IMBALANCE

For road segments matched to multiple hotspots, the final number of crashes associated to a road was calculated as the average of the number of crashes occurring in all hotspots matched to that road weighed by the inverse of its distance from the road.

We labeled the videos using an ordinal approach to avoid measurement error likely to be present in the crash data, which is a continuous variable. We classify the videos into 4 categories using a *k*-means algorithm. The number of clusters has been chosen in such a way to capture a decent level of danger heterogeneity but not letting *k* be too large. This led to a non-negligible imbalance in the class distribution, which was addressed in two different ways:

ORIGINAL CLASS DISTRIBUTION



Approach 1: Train with class imbalance but use a variety of evaluation metrics that partially account for class imbalance.
Approach 2: Balance classes in terms of inputs (number of video clips for each clip) by cropping shorter videos for categories 1 and 2 and longer videos for categories 3 and 4.

TRAINING AND TUNING

Each video is cropped to a sub-clip of pre-determined length (150s for all categories in Approach 1; 30s for categories 1 and 2, 75s for category 3 and 150s for category 4 in Approach 2) and divided into separate shorter video clips, our final inputs. For both Approach 1 and 2, video clips data set into train, validation and test using a proportion of 80:10:10. We trained 3 different types of models, at 1FPS and 10FPS. We used a learning rate of 0.0001 and mini-batch gradient descent with a batch size of 40 for 1FPS videos and 200 for 10FPS videos:

A-Models: A ConvNet-3D trained from scratch. Two 3D convolution layers (Conv1: 32 filters 5 x 5 x 5, Conv2: 64 filters 3 x 3 x 3) each followed by a 3D BatchNorm, ReLU and Dropout (*p=0.2*) layer. The convolution block is followed by a 3D MaxPool (2 x 2 x 2) and two fully connected (FC1: 256, FC2: 128) + ReLU layers, and a 4-class Softmax. Cross-Entropy Loss with Adam Optimizer.

B-Models: A pre-trained 18-layer ResNet-3D model. Two added hidden fully connected (FC1: 256, FC2: 128) + ReLU + Dropout (*p=0.2*) layers, and a 4-class Softmax. The ResNet-3D 18 model is pre-trained on Kinetics-400 dataset. During training, we freeze the weights of the ResNet 18 block, and train the two additional fully connected layers. Cross-Entropy Loss with Adam Optimizer.

C-Models: ConvNet-2D model based on ResNet-18 with random sampled single frames from each video-clip. Two fully connected (FC1: 256, FC2: 128) + ReLU + Dropout (*p=0.2*) layers, and a 4-class Softmax added to the pre-trained model and we train these layers only, keeping the ResNet weights fixed. Cross-Entropy Loss with Adam Optimizer with Adam Optimizer.

Tuned Hyperparameters: Clip Length (either 1.5s or 15s) + Number of Epochs (capped at 10).
Evaluation Metrics: Predicted Distribution of Labels + Aggregate and Class-Specific Accuracy (videoclip accuracy calculated by averaging the predicted probabilities from all the clips of a video).

RESULTS

CLASS DISTRIBUTIONS

	VALIDATION PREDICTED DISTRIBUTIONS			
Model	**Danger Level 1**	**Danger Level 2**	**Danger Level 3**	**Danger Level 4**
True Class Distribution (A1)	**38.46%**	**46.15%**	**7.69%**	**7.69%**
ConvNet-3D – 1FPS	7.69%	92.31%	0%	0%
ConvNet-3D – 10FPS	49.41%	48.23%	1.17%	0%
ResNet-3D – 1FPS	100%	0%	0%	0%
ResNet-3D – 10FPS (A1)	47.05%	47.05%	5.88%	0%
ConvNet-2D – 1FPS	88.46%	11.53%	0%	0%
ConvNet-2D – 10FPS	89.41%	10.58%	0%	0%
True Class Distribution (A2)	**40%**	**38.82%**	**12.94%**	**8.23%**
ResNet-3D – 10FPS (A2)	27.05%	36.47%	20%	16.47%

ACCURACY - APPROACH 1

	TRAINING ACCURACY				
Model	**Danger Level 1**	**Danger Level 2**	**Danger Level 3**	**Danger Level 4**	**Overall**
ConvNet-3D – 1FPS	72.54%	68.18%	0%	0%	57.26%
ConvNet-3D – 10FPS	89.21%	90.91%	36.66%	14.28%	78.63%
ResNet-3D – 1FPS	100%	2.27%	0%	0%	44.44%
ResNet-3D – 10FPS	90.19%	88.63%	60.00%	7.14%	80.76%
ConvNet-2D – 1FPS	89.21%	11.36%	0%	0%	43.16%
ConvNet-2D – 10FPS	98.03%	4.54%	0%	0%	44.44%

	VALIDATION ACCURACY				
Model	**Danger Level 1**	**Danger Level 2**	**Danger Level 3**	**Danger Level 4**	**Overall**
ConvNet-3D – 1FPS	20%	100%	0%	0%	53.84%
ConvNet-3D – 10FPS	64.70%	60.60%	0%	0%	49.41%
ResNet-3D – 1FPS	100%	0%	0%	0%	38.46%
ResNet-3D – 10FPS	55.88%	51.51%	18.18%	0%	44.70%
ConvNet-2D – 1FPS	100%	16.66%	0%	0%	46.15%
ConvNet-2D – 10FPS	100%	15.15%	0%	0%	45.88%

ACCURACY - APPROACH 2
(ResNet-3D – 10FPS only)

	ACCURACY				
Set	**Danger Level 1**	**Danger Level 2**	**Danger Level 3**	**Danger Level 4**	**Overall**
Training	64.22%	48.38%	66.27%	82.22%	59.41%
Validation + Test	52.94%	48.48%	36.36%	28.57%	47.05%

CONCLUSIONS AND FUTURE WORK

- Decent first-step results for a complex classification task.
- Need to improve performance. Different strategies to follow:
  - Try Deep 3D pre-trained models in which some of the earlier 3D layers are re-trained as well to account for the novelty of classification task.
  - For longer video-clips, possibly come up with a structure that reduces the loss of information due to averaging and pooling across the network before reaching the fully connected layers.
  - Explore the ordinality of our classes. During training, a true label belonging to category 1 should have higher mass of assigned predicted probability to categories 1 and 2 rather than 3 and 4 since 3 and 4 are increasingly more dangerous than 2.

MAIN REFERENCES

- Carreira, Joao, and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
- Chen, Quanjun, et al. "Learning deep representation from big and heterogeneous data for traffic accident inference." *Thirtieth AAAI Conference on Artificial Intelligence*. 2016.
- Hébert, Antoine, et al. "High-Resolution Road Vehicle Collision Prediction for the City of Montreal." *arXiv preprint arXiv:1905.08770* (2019).
- Yuan, Zhuoning, Xun Zhou, and Tianbao Yang. "Hetero-convlstm: A deep learning approach to traffic accident prediction on heterogeneous spatio-temporal data." *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018.