Deep Learning for Real-World Crime Recognition

Mart Rõbin, Artjom Geimanen, Anri Sokolov

Repository link: https://github.com/anriwv/CRIME-DETECTION

Task 2. Business Understanding

1. Identifying Business Goals

Background

Video surveillance is used almost everywhere today – in public places, shops, transport and many other locations. Normally, security operators have to watch these cameras manually, which is difficult and inefficient, especially when there are many cameras. People get tired and can easily miss important or dangerous situations.

Because of this, many organizations and researchers are interested in automatic activity recognition. The goal is not to replace humans completely, but to help them notice suspicious behavior earlier. For our project, we decided to explore if machine learning can classify actions in surveillance frames and identify which of them might be related to crime.

We are working with a public dataset that contains over one million 64x64 pixel frames extracted from videos and divided into 14 action classes.

Business Goals

- Understand whether machine-learning models can classify actions in low-resolution surveillance images.
- Build a simple system that can predict whether a frame shows criminal or non-criminal activity.
- Produce a working prototype and a documented process that can be reused or extended it future studies.

Business Success Criteria

- A classifier that reaches a reasonable accuracy.
- A classifier that can correctly identify crime-related frames with good recall. (missing a crime is worse than a false alarm)
- Evidence, that such a system could help reduce the workload of human operators.

2. Assessing the Situation

Inventory of resources

- Data: 1 266 345 training images and 111 308 test images, grouped into 14 classes.
- Hardware: Our own computers / Kaggle with free GPU
- Software: Python, PyTorch, TensorFlow, OpenCV, GitHub, Kaggle, Jupyter notebooks

Requirements, assumptions and constraints

- The dataset is too large to store directly in the GitHub repository, so we keep it locally.
- We assume that the dataset’s labels are correct and consistent.
- Since frames are only 64x64 pixels, many details are missing, which may limit model accuracy.

- Some activity classes may appear much more often than others.
- Training a deep neural network may take a long time on weak hardware.

Risks and contingencies

- Risk: Models might perform poorly due to low resolution.
  Plan: Start with smaller models and later try more advanced.
- Risk: Class imbalance may affect accuracy.
  Plan: Use class balancing methods.
- Risk: Training takes too long.
  Plan: Use smaller batch sizes or Kaggle/Google Colab GPUs if needed.
- Risk: Crime / non-crime definition may be unclear.
  Plan: Follow dataset documentation and keep our interpretation simple.

Terminology

- Frame – one image taken from a video.
- Activity class – one of the 14 action labels.
- Crime activity – a subset of classes that the dataset considers criminal.
- Binary classification – predicting crime or non-crime.
- Pipeline – the full process from data loading to evaluating the model.

Costs and benefits

- Cost: Time spent on exploring data, building models and writing documentation.
- Cost: Possible cost of cloud compute resources.
- Benefit: A basic system that shows how automatic crime detection might work.
- Benefit: Practical experience with large datasets and ML pipelines.
- Benefit: A foundation for future work on video or surveillance analysis.

3. Defining Data-Mining Goals

Data-Mining goals

- Build a model that predicts one of the 14 activity classes for each image.
- Build a second model that predicts crime or non-crime.
- Explore the dataset and understand class distribution, data quality and challenges.
- Document the modeling process according to CRISP-DM.

Data-Mining success criteria

- The models should reach clear, measurable performance goals (accuracy, recall, F1)
- The code and results should be reproducible by other people.
- The results should support the business goals defined earlier.

Task 3. Data understanding

1. Gathering data

Outline data requirements

- Video frames extracted from videos
- Uniform image format suitable for deep learning (same resolution and dimensions)
- Correct labels for each frame

Verify data availability

- 1 266 345 training images extracted from 1 610 videos
- 111 308 test images extracted from 290 videos
- All images represent frames extracted from ~ 128 hours of video footage, recorded at 30 fps.
- Every 10th frame from each video was extracted, new resolution 64x64 pixels

Datasets are available locally and through Kaggle.

Define selection criteria

All 14 classes will be used. Most images are suitable for model training. For first attempts of training may be used a test dataset or smaller train subset.

2. *Describing data*

General characteristics:

- Total images 1 377 653
- Total videos 1 900
- Number of classes 14
- Low image resolution 64x64

3. *Exploring data*

Label\ Dataset	Train	Test
Abuse	19 076	297
Arrest	26 397	3 365
Arson	24 421	2 793
Assault	10 360	2657
Burglary	39 504	7 657
Explosion	18 753	6 510
Fighting	24 684	1 231
Normal Videos	947 768	64 952
Road Accidents	28 486	2 663
Robbery	41 493	835
Shooting	7 140	7 630
Shoplifting	24 835	7 623
Stealing	44 802	1 984
Vandalism	13 626	1 111

- Wide range in image quantity. Non-crime actions dominate in the dataset.
- “Assault” and “Shooting” have very few samples.
- Some images contain motion blur.

- Some frames contain empty or black screens.
- Black bars appear on the top and bottom of some frames due to aspect-ratio differences in the original source videos.

4. *Verifying data quality*

Completeness

- All images have valid labels, there are no unlabeled ones.
- All frames load correctly, and image dimensions are consistent.

Consistency

- All images share the same format.
- Label structure is the same across training and test sets.

Accuracy

- Some frames contain motion blur
- Black frames appear in a small portion of data
- Black bars in videos.

A critical issue is that the dataset does not contain frames explicitly labeled as crime or frames where crime starts.

One class contains 70% of all frames.

Task 4. Planning your project

Our project focuses mainly on model development, experimentation, and evaluation. Only manual step required is collecting images into the correct series array. Each team member contributes equally across tasks

		Estimated
Task	Member	hours
Data processing: data loaders, train/validation split, preparing series arrays	Mart	2
	Artjom	2
	Anri	2
Testing pretrained models (EfficientNet, MobileNet, ResNet)	Mart	2
	Artjom	2
	Anri	2
Research	Mart	3
	Artjom	3
	Anri	3
Training models using transfer learning for 14 classes	Mart	3
	Artjom	3
	Anri	3
Comparing results	Mart	1
	Artjom	1
	Anri	1

Designing and training a custom architecture	Mart	5
	Artjom	5
	Anri	5
Comparing results	Mart	1
	Artjom	1
	Anri	1
Hyperparameter tuning	Mart	2
	Artjom	2
	Anri	2
Retraining with optimized parameters	Mart	5
	Artjom	5
	Anri	5
Selecting the best model using accuracy, recall, and F1-score	Mart	1
	Artjom	1
	Anri	1
Develop a small demo application to showcase the model’s predictions.	Mart	5
	Artjom	5
	Anri	5