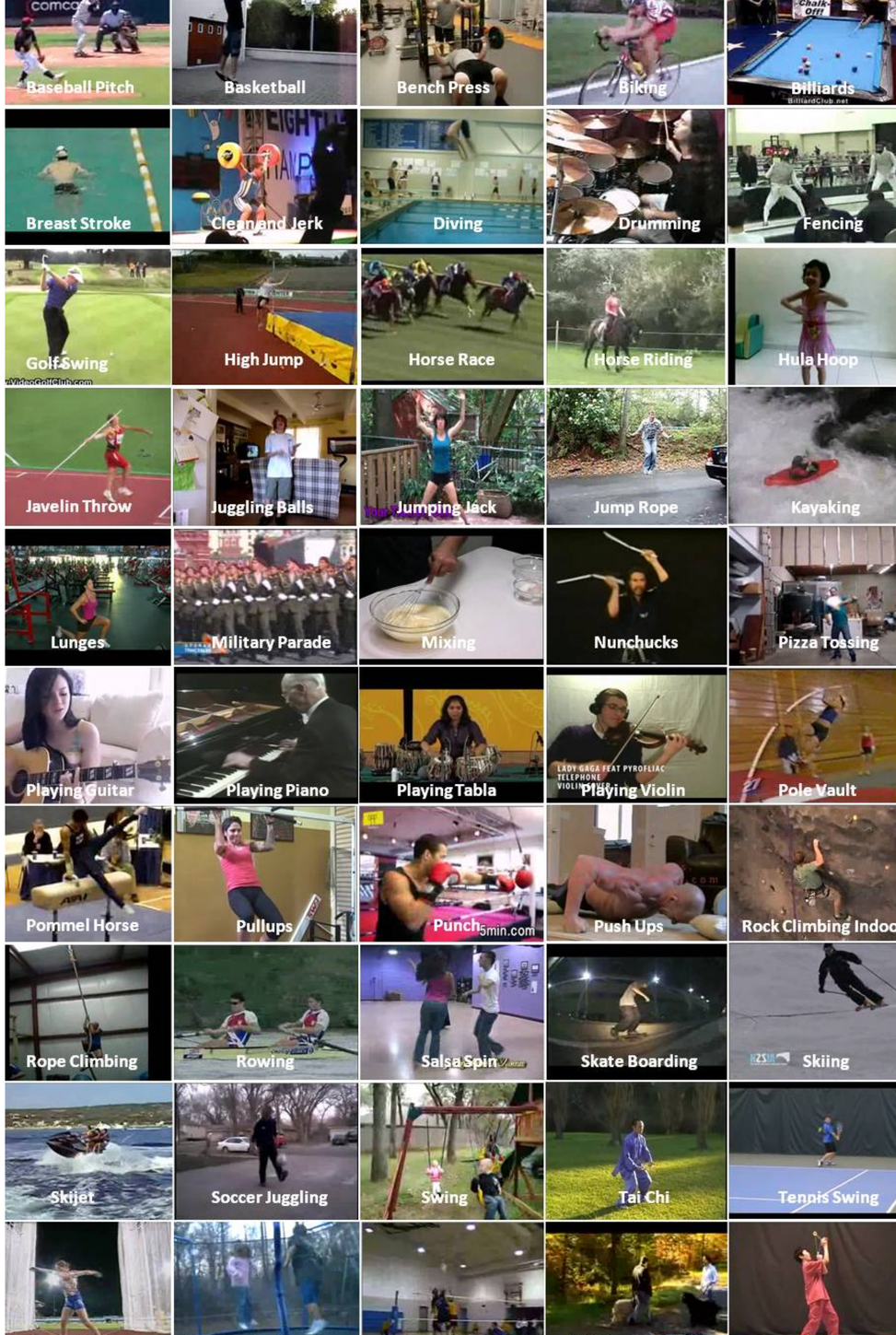
Enhancing Action Recognition & Advanced Frames Selection

Innopolis University, Computer Vision Course 2024



Project Overview

Problem Statement:

Video action recognition is computationally demanding and resource-intensive.

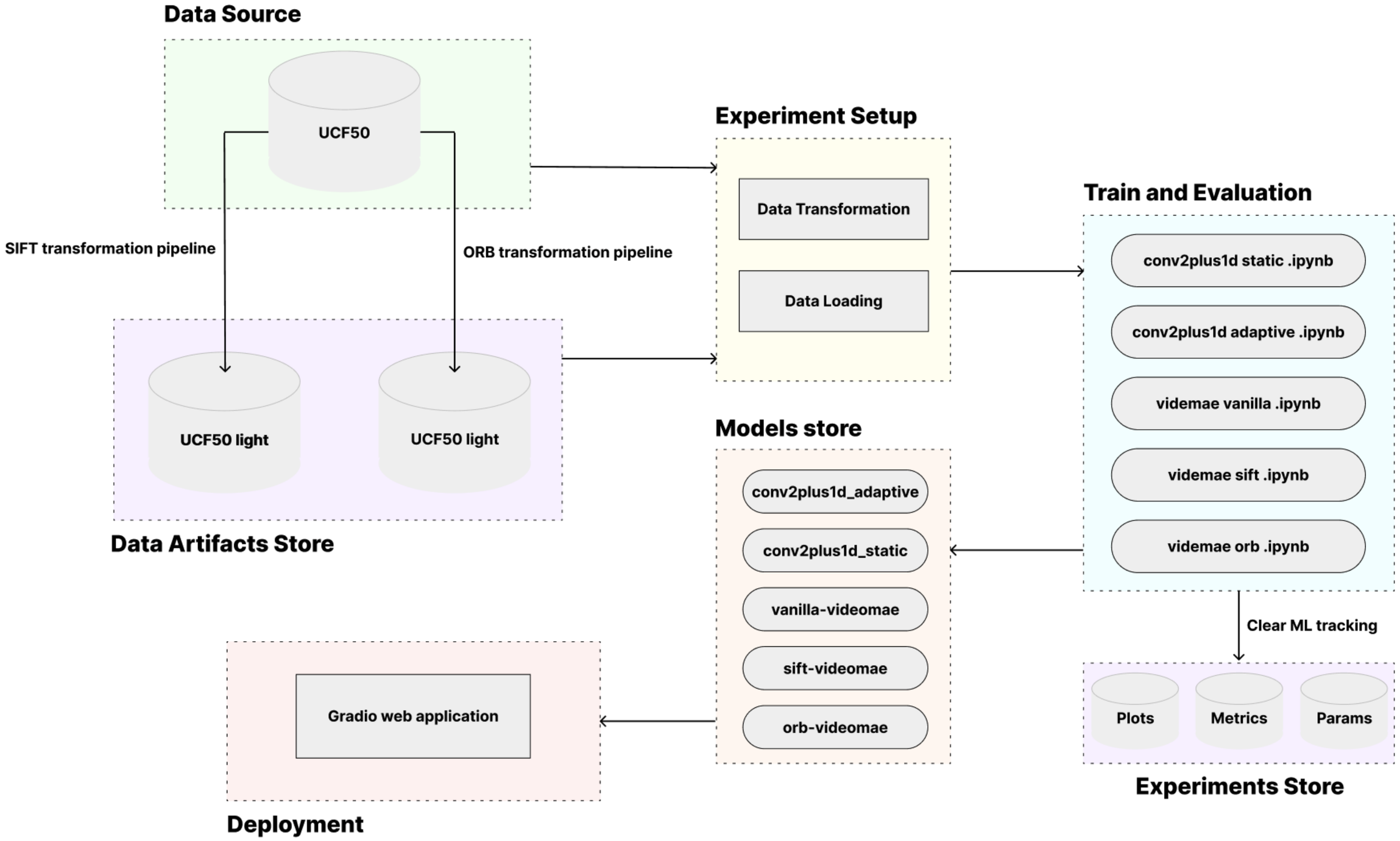
Objective:

Develop a lightweight video action recognition pipeline that reduces data size while maintaining competitive accuracy.

Key Performance Indicators:

Reduction in data size, training and inference time, and maintenance of classification metrics (F1, accuracy, precision, recall, ROC-AUC).

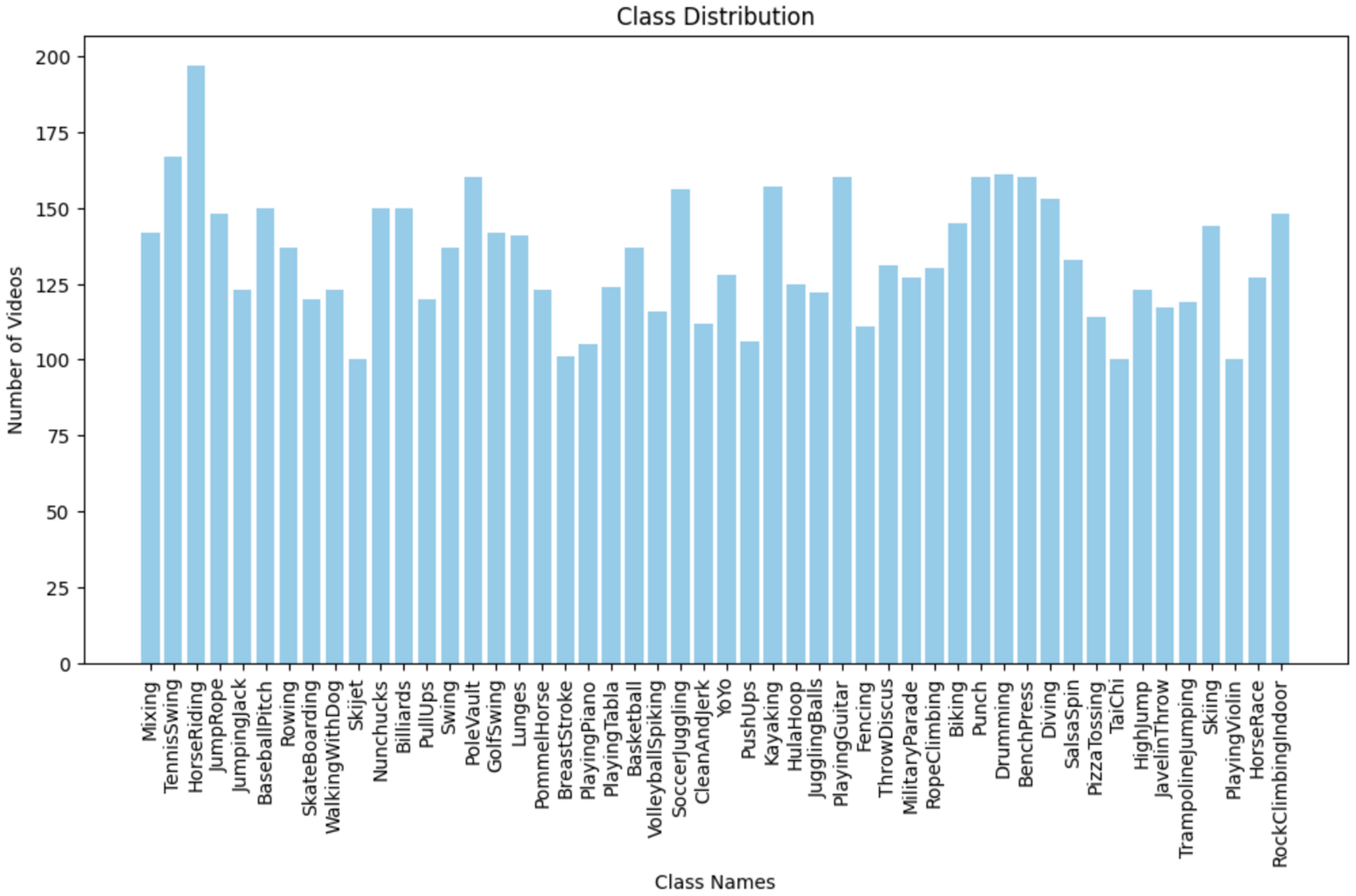
Project architecture


Data Source
UCF50
SIFT transformation pipeline
ORB transformation pipeline
UCF50 light
UCF50 light
Data Artifacts Store
Experiment Setup
Data Transformation
Data Loading
Train and Evaluation
conv2plus1d static .ipynb
conv2plus1d adaptive .ipynb
videmae vanilla .ipynb
videmae sift .ipynb
videmae orb .ipynb
Models store
conv2plus1d_adaptive
conv2plus1d_static
vanilla-videomae
sift-videomae
orb-videomae
Clear ML tracking
Plots
Metrics
Params
Experiments Store
Gradio web application
Deployment

Research design

UCF 50 Dataset

6,618 clips **50 categories** **3.23 GB size**

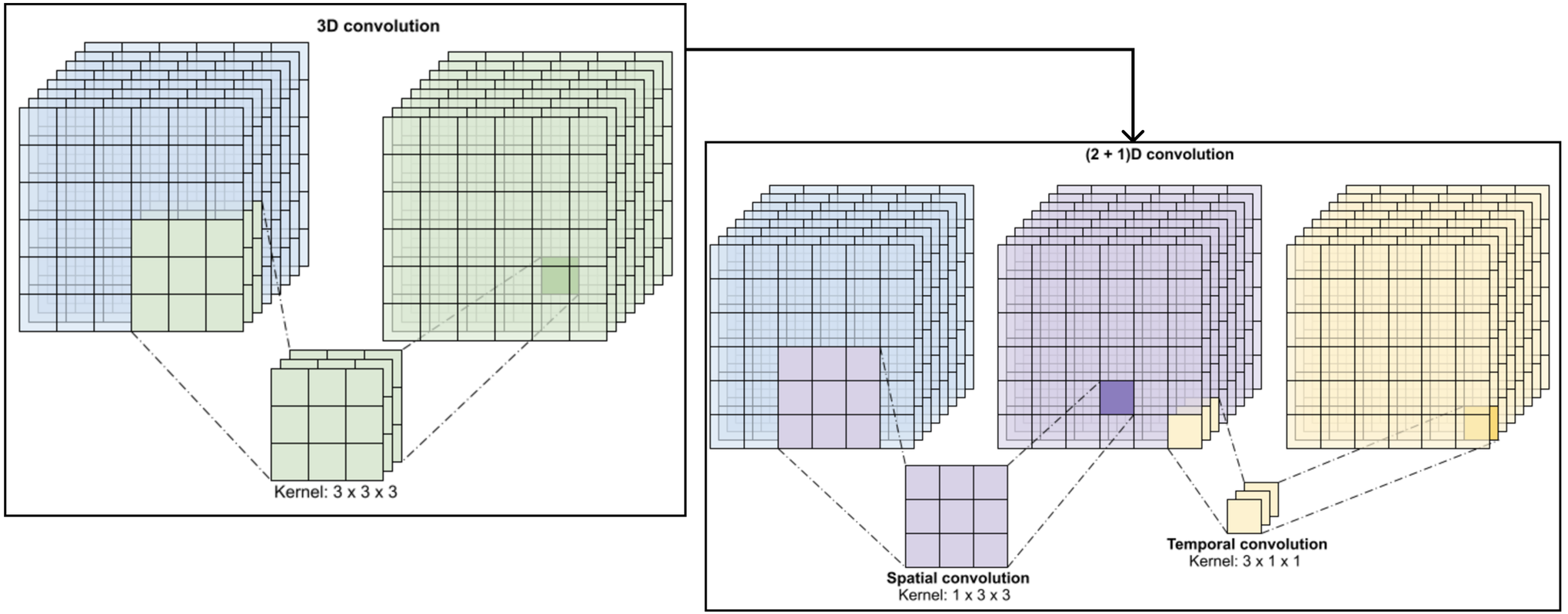


Models

(2+1)D Convolutions

Basic idea

param#3D = (3*3*3)*in*out+out



param#(2+1)D = (1*3*3)*in*out+out +
+ (3*1*1)*out*out+out

Article: https://arxiv.org/abs/1711.11248

Models

(2+1)D Convolutions

Results

Static Method: step_frame = const

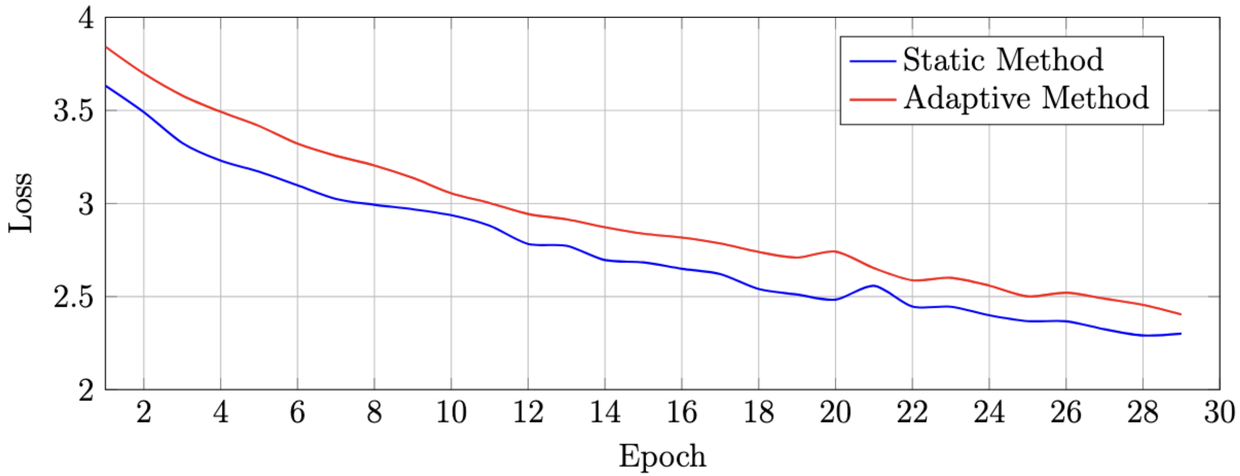
Adaptive Method: step_frame = video_length/n_frames

Method	Test Loss	Test Accuracy	Macro-average ROC-AUC
Static Method	2.288	0.376	0.93
Adaptive Method	2.401	0.364	0.92

Models

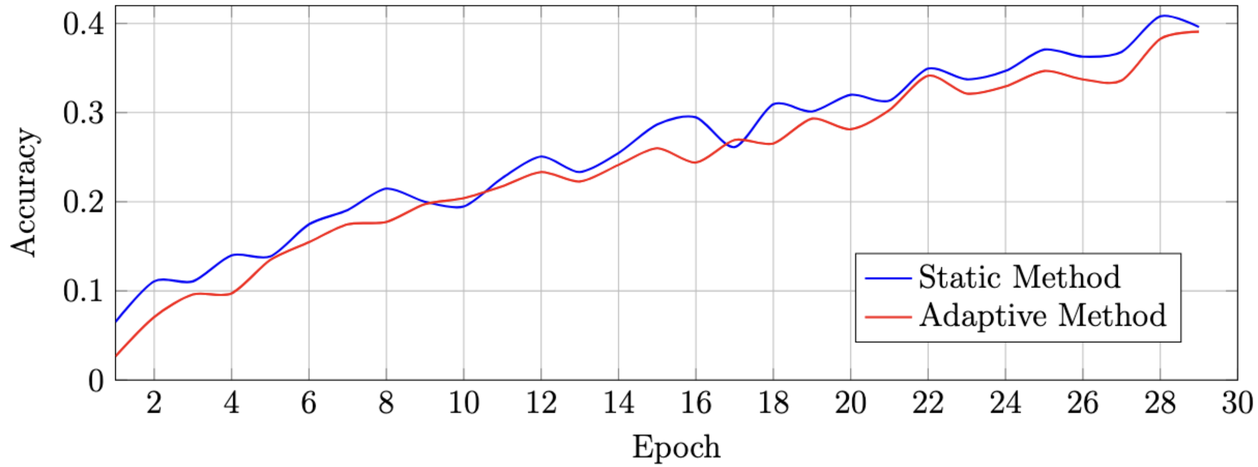
(2+1)D Convolutions

Results



⟵ Validation loss

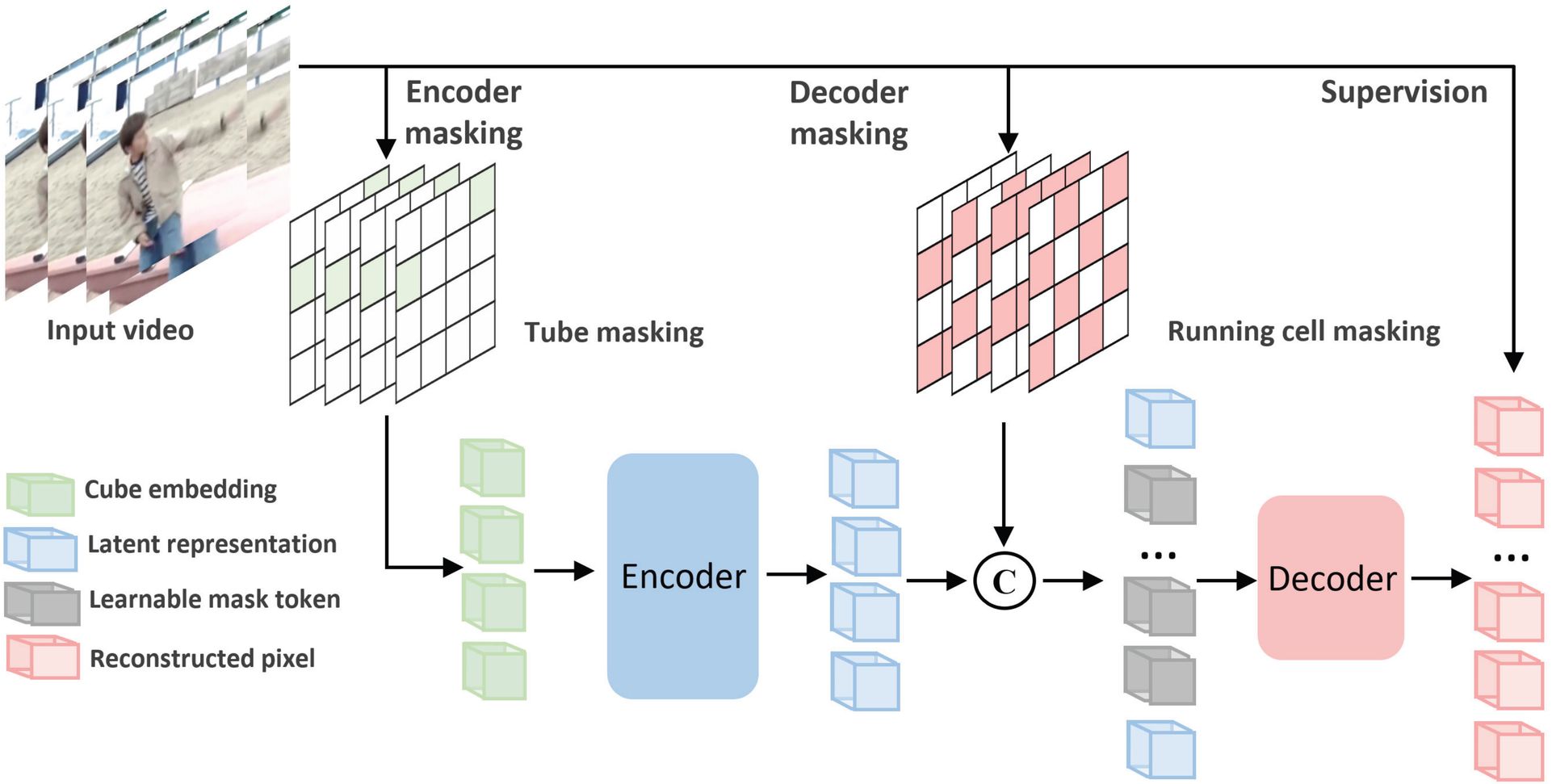
Validation accuracy ⟶



Models

VideoMAE

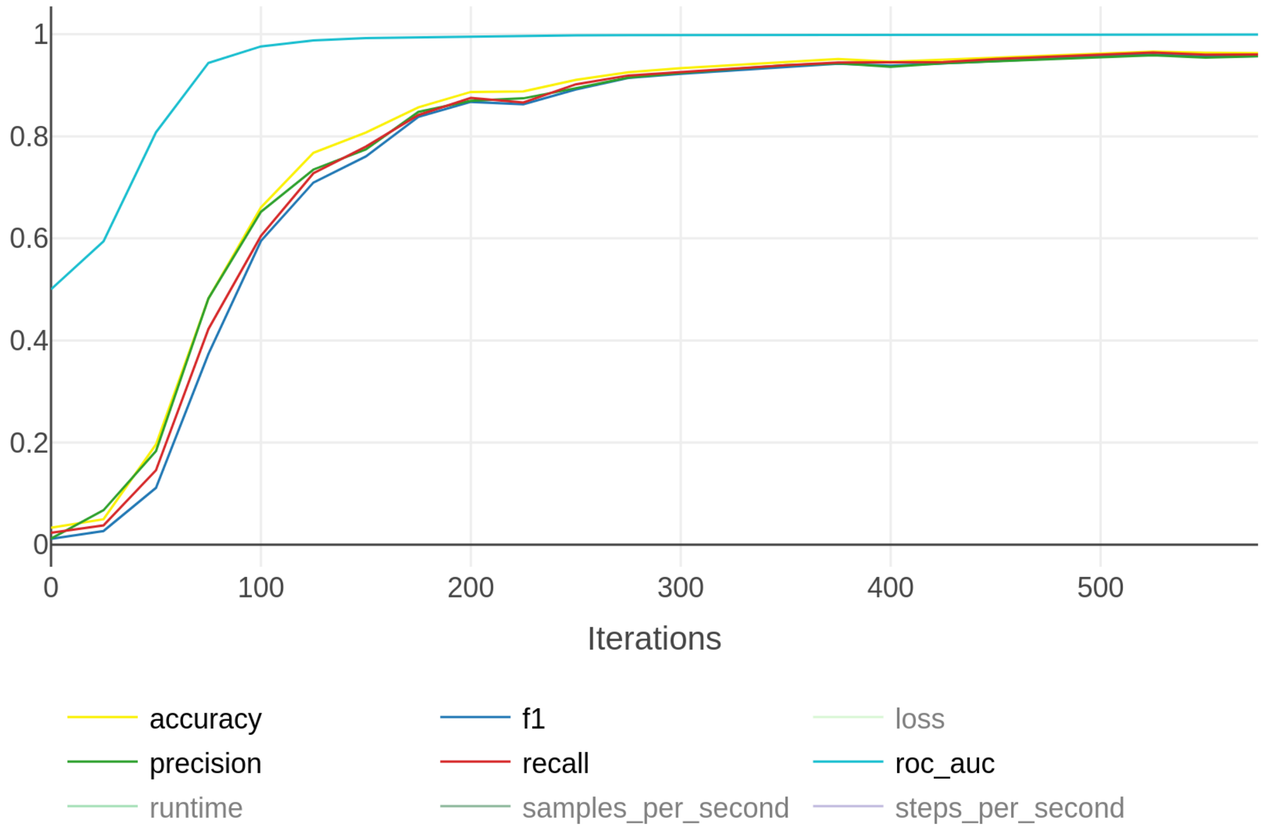
Basic idea



Models

VideoMAE

Results



Model Comparison

Models

	Parameters	Size (.safetensors)	Pretrained	Framework
(2+1)D Convolutions	0.4M	1.8MB	-	
VideoMAE	94.2M	330MB	Kinetics-400	

Frames Selection

ORB

(Oriented FAST and Rotated BRIEF)

Identify keypoints and compute their descriptors

```
keypoints, descriptors = orb.detectAndCompute(processed_frame, None)
```

Compare descriptors using Hamming distance

```
bf = cv2.BFMatcher(cv2.NORM_HAMMING, crossCheck=True)
matches = bf.match(prev_descriptors, descriptors)
```

```
distances = [m.distance for m in matches]
avg_distance = np.mean(distances)
```

Calculate distances

```
if avg_distance > threshold:
    relevant_frames.append(frame)
```

Select frames based on a distance threshold

Frames Selection

SIFT

Scale-Invariant Feature Transform

Identify keypoints and compute their descriptors

```
keypoints, descriptors = sift.detectAndCompute(processed_frame, None)
```

Find potential matches

```
bf = cv2.BFMatcher()
matches = bf.knnMatch(prev_descriptors, descriptors, k=2)
```

Apply Lowe's ratio test to filter matches

```
good_matches = []
for match in matches:
    if len(match) == 2:
        m, n = match
        if m.distance < 0.75 * n.distance:
            good_matches.append(m)
```

Maintain a record of matches

```
total_frames += 1
all_frames.append((total_frames, len(good_matches), processed_frame))
```

Frames Selection

SIFT

Scale-Invariant Feature Transform

Sort all frames by the number of good matches (ascending)

```python
all_frames.sort(key=lambda x: x[1])
```

Choose the top frames

```python
selected_frames = all_frames[:min(len(all_frames), 2 * number_of_frames)]
```

Reorder the selected frames to match their original sequence

```python
selected_frames.sort(key=lambda x: x[0])
```

Save only the contents of the frame

```python
relevant_frames = [frame[2] for frame in selected_frames]
```

Dataset Size Comparison

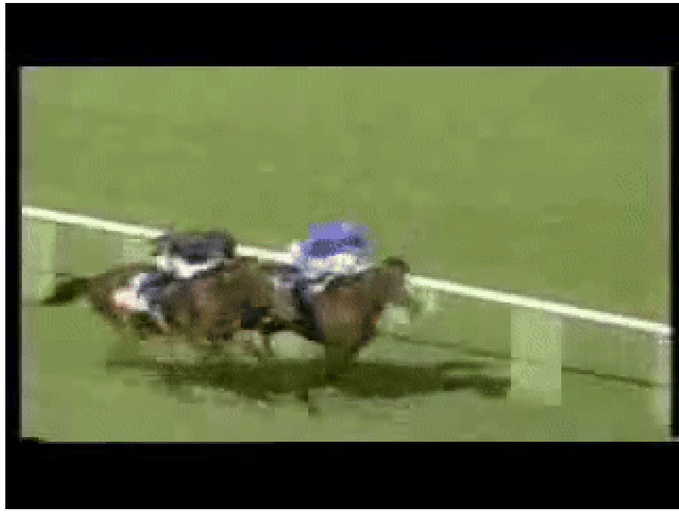
Dataset	UCF50	UCF50 ORB	UCF50 SIFT
Size of zip file	3.23 GB	0,87 GB	1.62 GB

Results

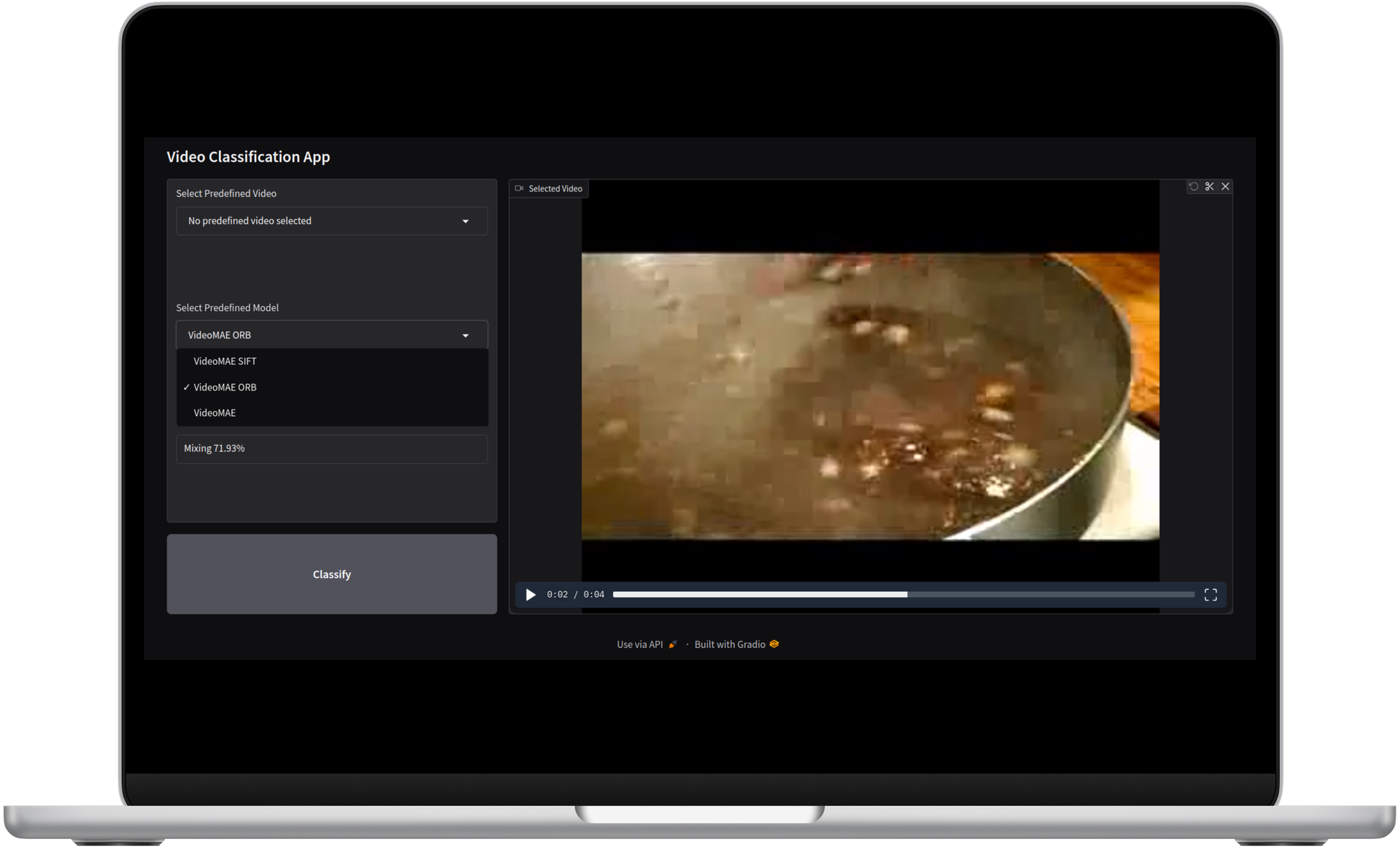
Original

ORB

SIFT

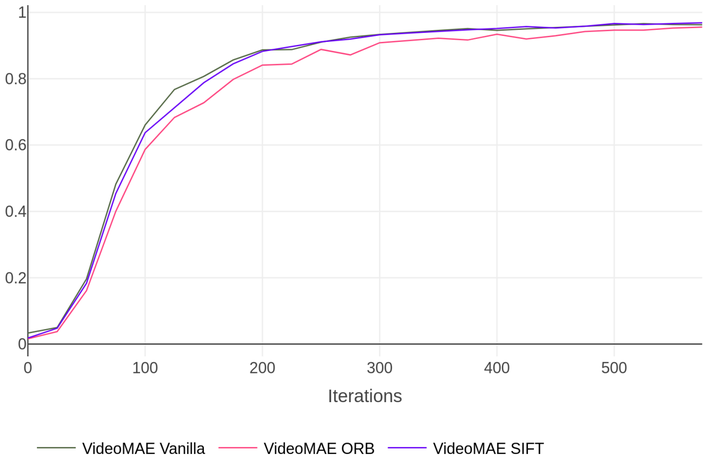


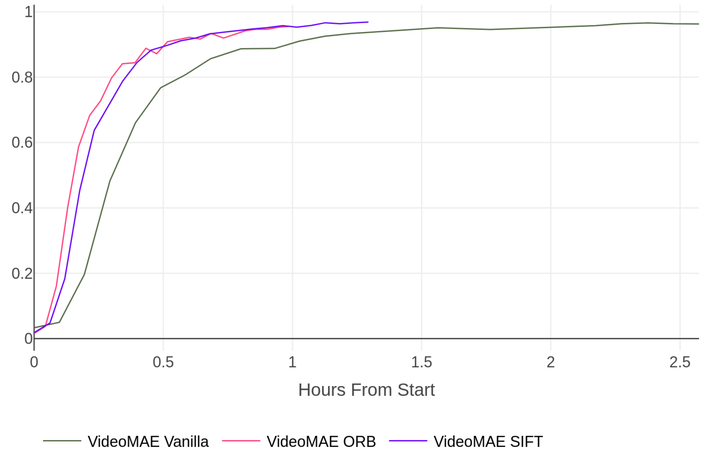
Let’s see the demo



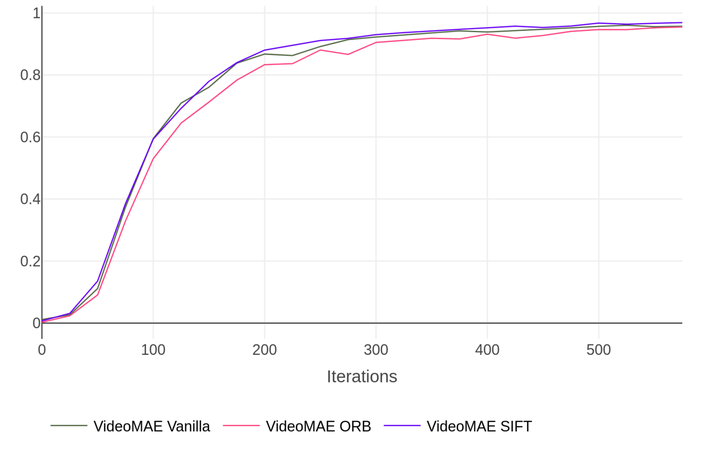
Results

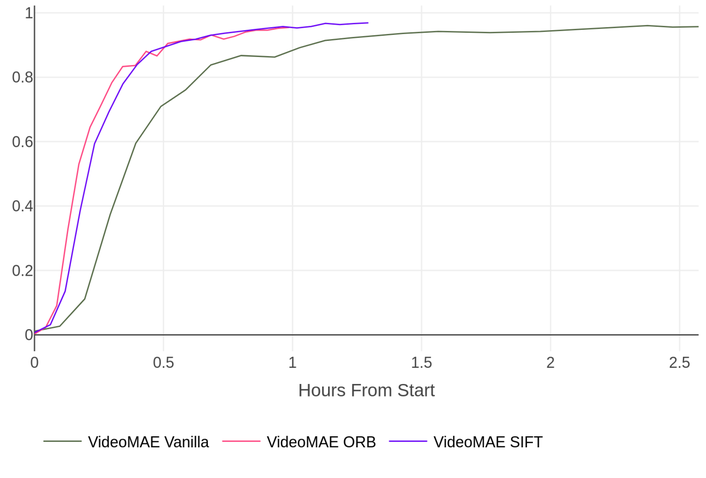
VideoMAE Accuracy Comparison





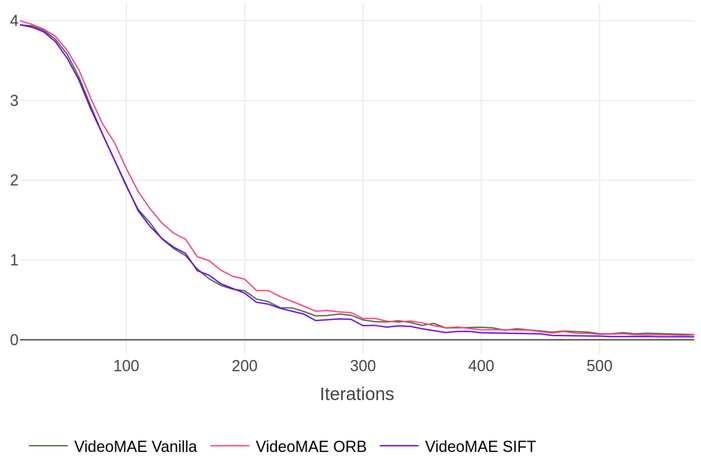
VideoMAE F1 Comparison

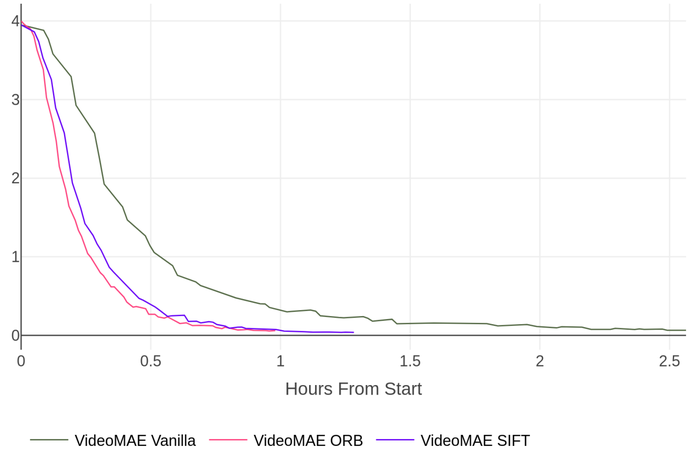




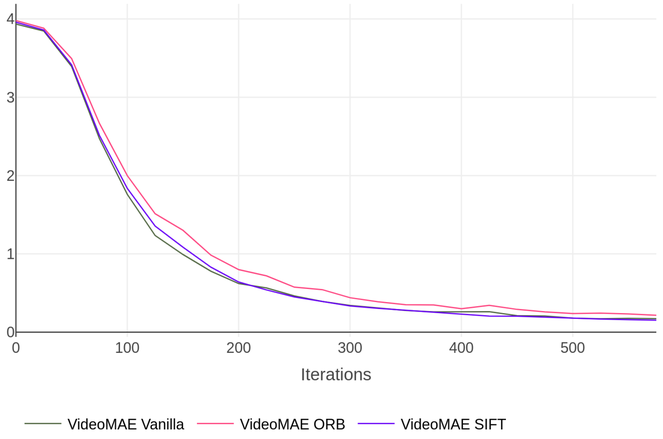
Results

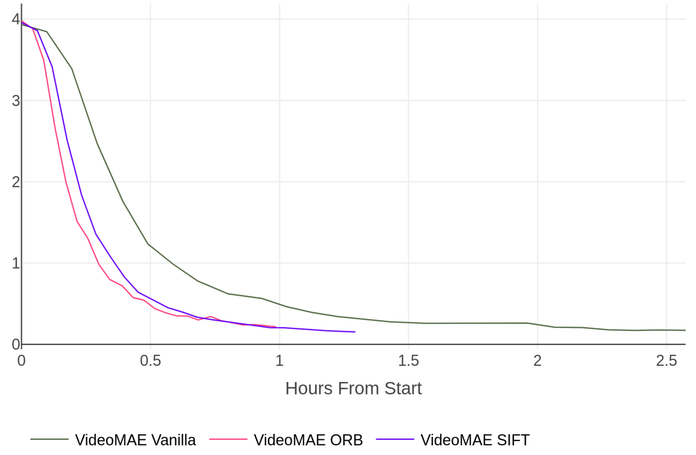
VideoMAE Train Loss





VideoMAE Eval Loss





Results

Key Idea: selecting the most informative frames reduces dataset size and processing time without significantly degrading performance.

Findings

- A model with reduced dataset achieves almost the same accuracy as a model with the full dataset.
- Models reach optimal performance faster, reducing training time.
- Less computational resources (memory, processing time) are used.

Project Team

Ivan Golov

Teamlead, A100 hunter, ORB master

Anatoly Soldatov

Transformer enjoyer, MLOps conqueror

Rufina Gafiiatullina

Convolutional majesty, SIFT guru

Project timeline

Week 6

Data collection and research

Week 7-8

EDA and POC

Week 9-10

Data transformation pipeline, models traning and evaluation

Week 11-12

Comparative analysis of observations

Week 13

Project presentation

Q/A section