AMVP: Adaptive CNN-based Multitask Video Processing on Mobile Stream Processing Platforms

Mengyuan Chao, Radu Stoleru, Liyi Jin, Shuochao Yao*, Maxwell Maurice†, Roger Blalock†
Department of Computer Science and Engineering, Texas A&M University
*Department of Computer Science, George Mason University
†National Institute of Standards and Technology (NIST)

The Fifth ACM/IEEE Symposium on Edge Computing (SEC’20)
Virtual, November 11-13, 2020
Outline

• Background & Motivation

• Related Work

• System Overview

• Adaptive Multitask Video Processing Framework

• Experimental Results

• Conclusion & Future Work
Background – Effective Disaster Response

Effective Disaster Response (EDR)

Challenges

Edge Computing Communication (ECC)

Pictures source: http://www.stayinbusiness.com/
Background – Techniques for ECC Enabling EDR

- **Current: System On Wheels (SOW)**

  - Incident Response Vehicle
  - AWS Public Safety & Disaster Response

  - Limitations of SOW
    - Need long time to arrive
    - Cannot go harshest area
    - Expensive to equip with

- **Next Generation: Edge Bubble (EB)**

  - Manpack
  - eNodeB
  - EPC
  - WiFi router

  - Video App
  - Logging App
  - Voice App

  - First responders work as a team

- **Advantages of EB**
  - Shorter time to arrive
  - Can reach harshest area
  - Cheaper to equip with
  - Closer to data source

Picture source: [https://www.fema.gov](https://www.fema.gov), Federal Emergency Management Agency

Motivation – Multitask Video Processing at EB

- **Scenario**: First responders need automatically record multiple information about survivors

  - Old or young?
  - Male or female?
  - Looks OK or not?

- **Challenges**
  - Limited computing resources on mobile devices
  - Computation intensive video stream analysis
  - Dynamic computing resources and networks
  - Diverse user performance requirements

- **Idea**: Combine multiple pipelines into a single one and adaptively share and offload CNN layers

Related Work – CNN-based Mobile Vision Processing

- **CNN Offloading**
  - **Cloud**: DDNN@ICDCS’17, Neurosurgeon@SIGARCH’17, JALAD@ICPADS’18, µLayer@EuroSys’19
  - **Cloudlet**: MODI@HotEdge’18, IONN@SoCC’18, DADS@Infocom’19, Couper@SEC’19
  - **IoT/Mobile**: Modnn@DATE’17, Mednn@ICCAD’17, MusicalChair@arXiv’18, DeepThings@TCAD’18

- **CNN Compression**
  - **One-fit-all**: XNOR-Net@ECCV’16, Thinet@ICCV’17, FactorizedCNN@ICCV’17, ShuffleNet@CVPR’18
  - **Adaptive**: DeepX@IPSN’16, AdaptDNN@LCTES’18, OnDemandDNN@MobiSys’18, ContextDNN@ICDCS’20
  - **Feature**: DeepFCPX@ICIP’18, ContextFCPX@CVPR’18, LosslessFCPX@MMSP’18, LossyFCPX@ICM’19

- **CNN Sharing**
  - **Inside a Model**: NestDNN@MobiCom’18, FoggyCache@MobiCom’18, DeepCache@MobiCom’18
  - **Among Models**: MCDNN@MobiSys’16, Mainstream@ATC’18

We study adaptive CNN-based multitask video processing on mobile devices with dynamic computing resources, network, and user goals.
System Overview – Hardware Architecture

Hardware Implementation of Edge Bubble

- WiFi/LTE manpack
- Helmet camera
- Carry-on mobile devices
System Overview – Software Architecture

- **AMVP**: Adaptive CNN-based Multitask Video Processing Framework

![Diagram of AMVP software architecture]

**Software Architecture of AMVP**

- **Model Training**: Pre-trained models
  - Classifier replacing
  - Layer freezing
  - Model re-training
  - Re-trained models

- **Model Splitting**: Complete .h5 models
  - Model splitting
  - Model converting
  - Splitted .tflite models

- **Model Profiling**: Latency profiler (L)
  - Memory profiler (M)
  - Accuracy profiler (A)
  - Traffic profiler (T)

- **Model Selection & Task Assignment**
  - Runtime resource & network monitor
  - App topology & user preference Manager
  - Optimization Function
    - Optimal model selection
  - Assignment Schemes
    - Optimal task assignment

**Offline**

**Online**

-pre-trained models: e.g., mobileNetV2
-vision datasets: e.g., emotion dataset

**T**

**S**

**P**

**S&A**
Adaptive Multitask Video Processing Framework

- **Model Training via Transfer learning**
  - **Strategy 1**: retrains all weights, requires a large dataset and a lot of computation
  - **Strategy 2**: freezes whole convolutional base, only trains classifier; suitable for training tasks similar to original task
  - **Strategy 3**: freezes some layers in base, trains the rest; a small dataset, freezes more; a large dataset, retrains more

- **Transfer learning results**
  - **Observation 1**: resNet50V2-based are more accurate than mobileNetV2-based
  - **Observation 2**: Strategy 3 achieves optimal performance for our tasks and datasets
  - **Observation 3**: Simpler task is less affected by # frozen layers than complicated tasks
Adaptive Multitask Video Processing Framework

- Model Profiling

- Observation 1: latency of P1 increases with block number
- Observation 2: memory is concentrated in the last few layers
- Observation 3: feature size gradually gets reduced along with inference process
Adaptive Multitask Video Processing Framework

- Model Splitting

<table>
<thead>
<tr>
<th>Split at earlier layer</th>
<th>VS.</th>
<th>Split at latter layer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Higher inference accuracy</td>
<td>➢</td>
<td>Share more among CNNs</td>
</tr>
<tr>
<td>Better computation balance</td>
<td>➢</td>
<td>Better memory balance</td>
</tr>
<tr>
<td>Share less among CNNs</td>
<td>➢</td>
<td>Smaller feature traffic size</td>
</tr>
<tr>
<td>Worse memory balance</td>
<td>➢</td>
<td>Lower inference accuracy</td>
</tr>
<tr>
<td>Larger feature traffic size</td>
<td>➢</td>
<td>Worse computation balance</td>
</tr>
</tbody>
</table>

- Quantization-based Feature Compression

\[
\tilde{F} = \left[ \frac{F - \min(F)}{\max(F) - \min(F)} \cdot (2^n - 1) \right] \\
\hat{F} = \frac{\max(F) - \min(F)}{2^n - 1} \cdot \tilde{F} + \min(F)
\]

32 bits \(\rightarrow\) \(n\) bits

![Diagram showing model splitting and quantization-based feature compression](https://lenss.cse.tamu.edu)
Adaptive Multitask Video Processing Framework

- Model Selection and Task Assignment (MSTA)
  - Cost function for each task
    \[ C(m_s, u^k_s, s) = \alpha_s \cdot \frac{A_s - A(m_s)}{A_s} + \beta_s \cdot \frac{\max(0, L(m_s, u^k_s) - L_s)}{L(m_s, u^k_s)} + \gamma_s \cdot \frac{\max(0, T_s - T(m_s, u^k_s))}{T_s} \]
    \[ L(m_s, u^k_s) = \frac{1}{u^k_{m_1} u^k_{m_2}} + \lambda_s \frac{f_{k_1} m_1^2}{u^k_{m_1} m_1^2} + \lambda_s \frac{f_{k_2} m_2^2}{u^k_{m_2} m_2^2} + D(f_{k_1} m_1^2, b_{k_1 k_2}) \]
  - Problem formulation
    \[ \text{minimize} \quad C \]
    \[ \text{subject to:} \quad \forall s : C(m_s, u^k_s, s) \leq C, \]
    \[ \forall k : \sum_{\{m^i_s\}} u^k_{m^i_s} \leq U_k, \]
    \[ \forall k : \sum_{\{m^i_s\}} r^k_{m^i_s} \leq R_k \]
Adaptive Multitask Video Processing Framework

➢ Greedy algorithm for MSTA

(a) Basic workflow of MSTA (b) An example of MSTA
Experimental Results

- Adapt to different accuracy requirements
- Adapt to different latency requirements
Experimental Results

- Adapt to different computing resources
- Adapt to different network conditions
Conclusion and Future Work

• **Conclusion**: We propose **AMVP**, an Adaptive Multitask Video Processing Framework which supports **dynamic CNN layer sharing** among multiple CNN-based vision analysis tasks and adaptive CNN layer offloading from one mobile device to other devices at an **Edge Bubble**.

• **Future Work**

  ➢ Apply accuracy metrics as in Chameleon [Sigcom’18], which includes precision, recall and F1 score.
  ➢ Generalize AMVP to deploy on a heterogeneous stream processing platform including edge server
  ➢ Support simultaneous processing of multiple videos
  ➢ Support other AI applications such as voice recognition, speech recognition, NLP, etc.
Thank you!

Q&A

Contact: stoleru@cse.tamu.edu