Design Project Presentation

Topic: Video Processing

3rd May 2023

PRESENTED BY

Mayank Singh Rajput (B19CSE054)
Mohit Ahirwar (B19CSE055)
Ram Khandelwal (B19CSE116)

PROJECT MENTOR

Dr. Binod Kumar

Agenda

Section 1

Video Segmentation

Section 2

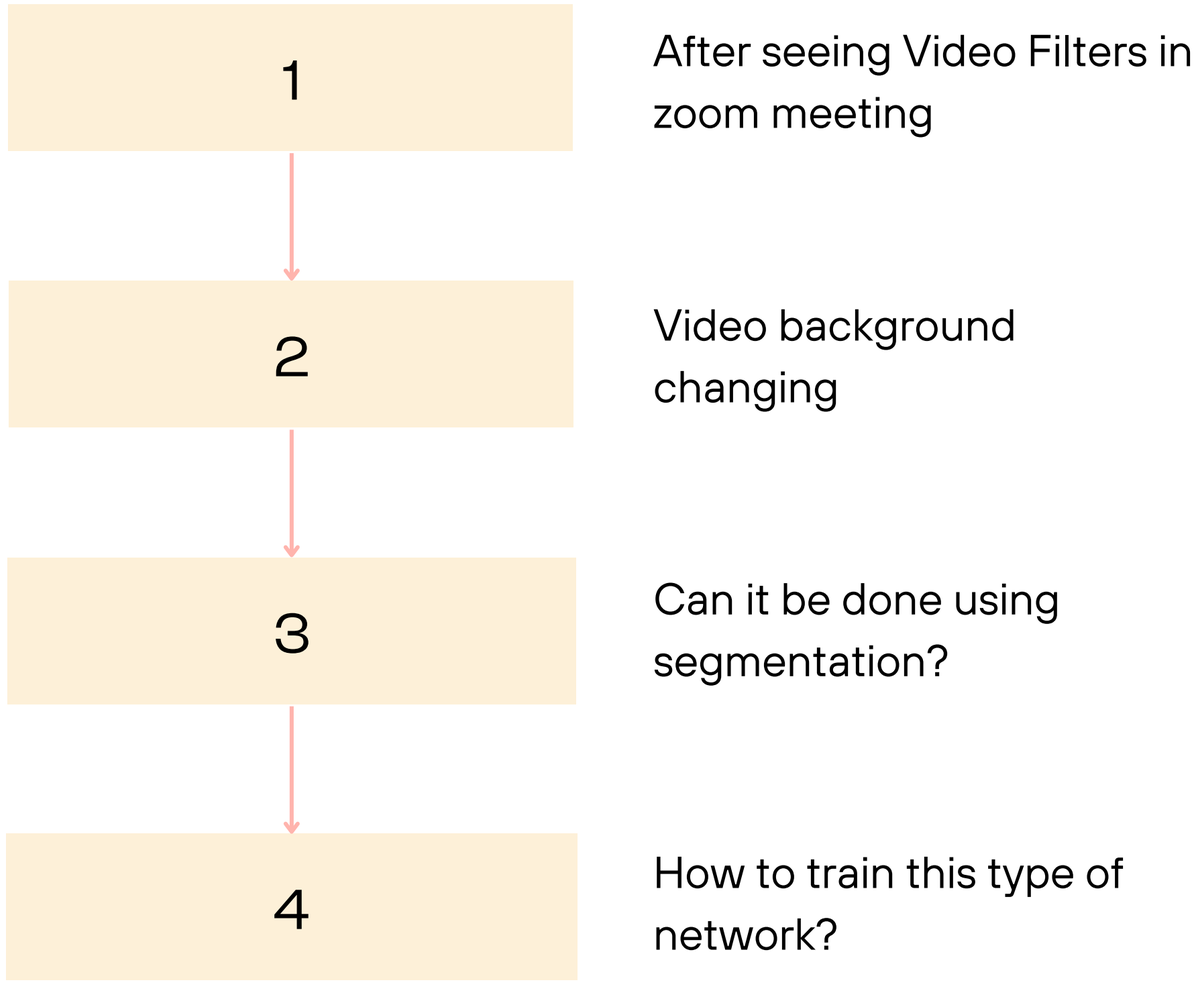
Video Compression

Video Segmentation

Let's first discuss about video segmentation

Motivation

Here are some of the use cases and ideas to think about while thinking of video segmenting



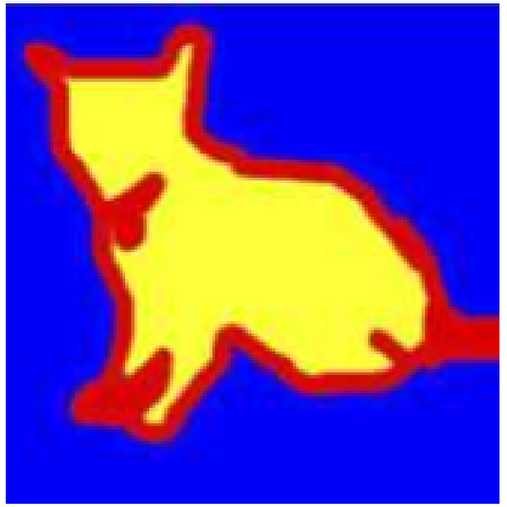
Dataset

We used Oxford-IIIT pet dataset.

- The Oxford-IIIT pet dataset is a 37 category pet image dataset with roughly 200 images for each class. The images have large variations in scale, pose and lighting.
- All images have an associated ground truth annotation of breed.
- Available with tensorflow dataset

Objectives

Let's see what are the objectives here



PET IMAGE

SEGMENTED OUTPUT

Preprocessing the Data

Steps involved for preprocessing

- Data Augmentation
- Uniform flipping
    - Input image and its ground truth label
- Resizing images to (128*128)
- Normalizing image

Splitting Data

Here are the steps to split the data

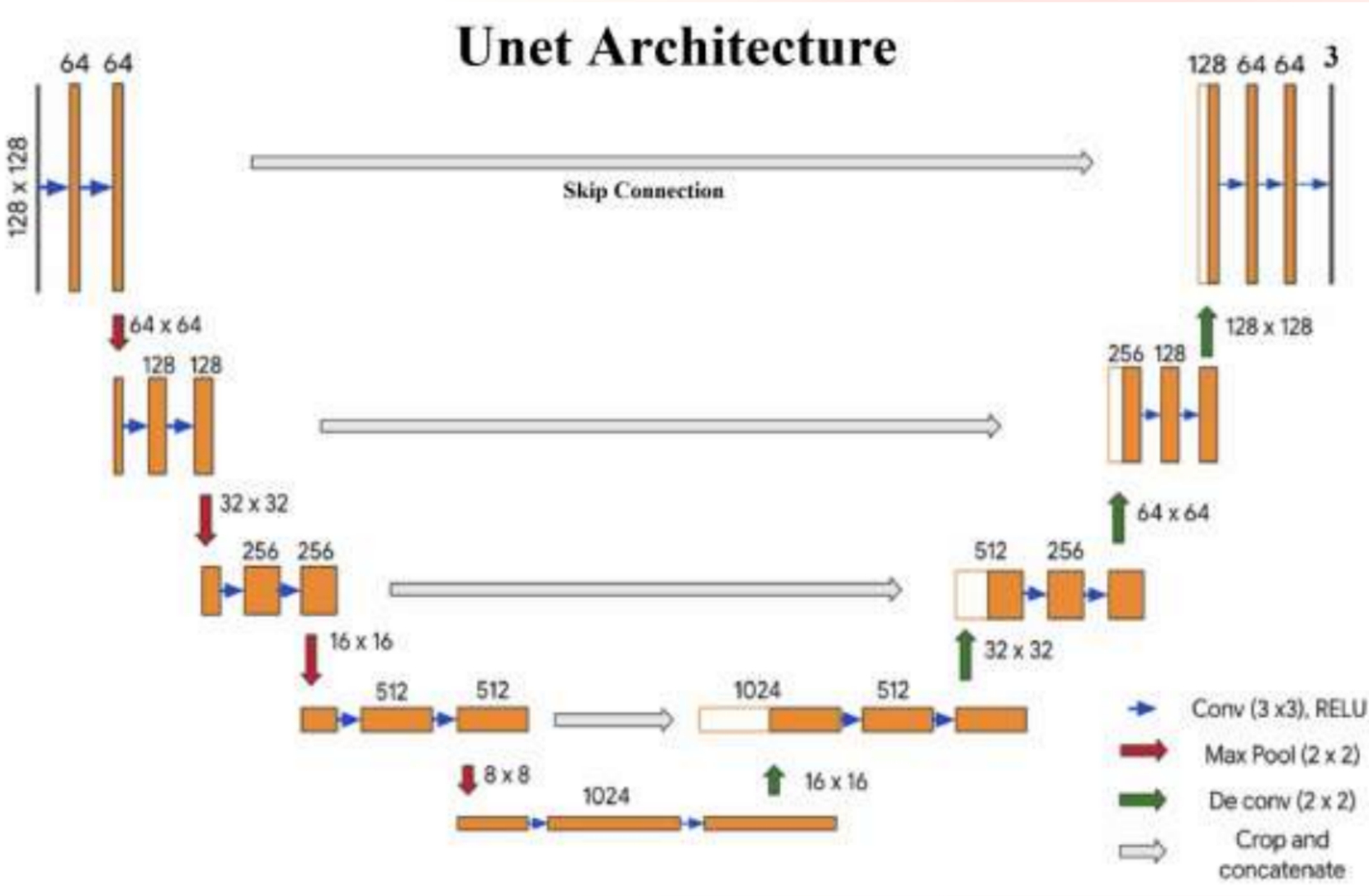
- As it is supervised learning
- Data- train test split
- We train the model on some data and test the model o remaining data

Segmentation using U-net Architecture

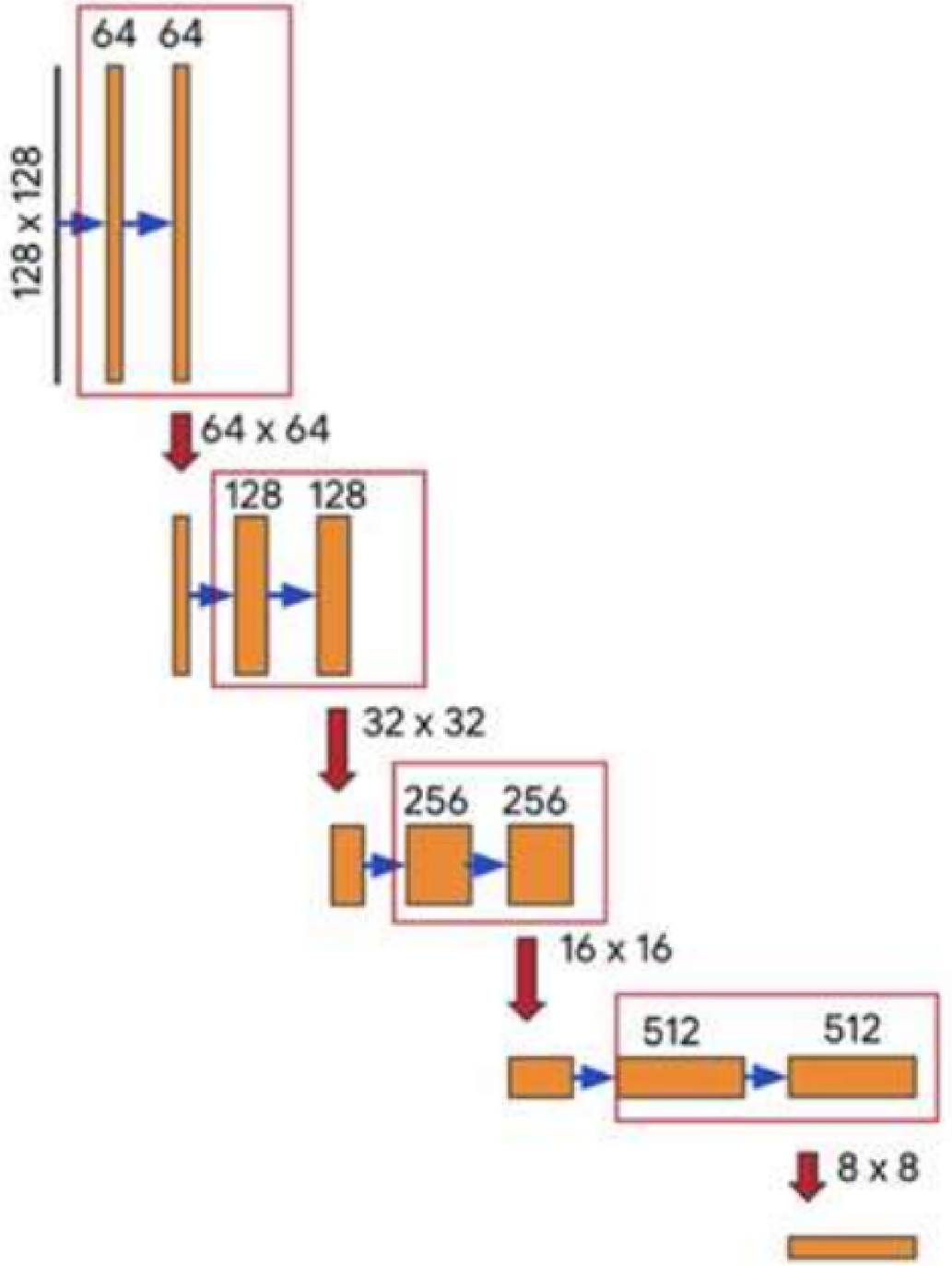
Using u-net architecture for segmenting

- Segmentation – pixel wise classification
  - Clustering of pixels which share same property.
- U-net architecture is two stage process
  - Down sampling and Up sampling
- Sampling is used to extract features which helps to classify each pixel to corresponding class.

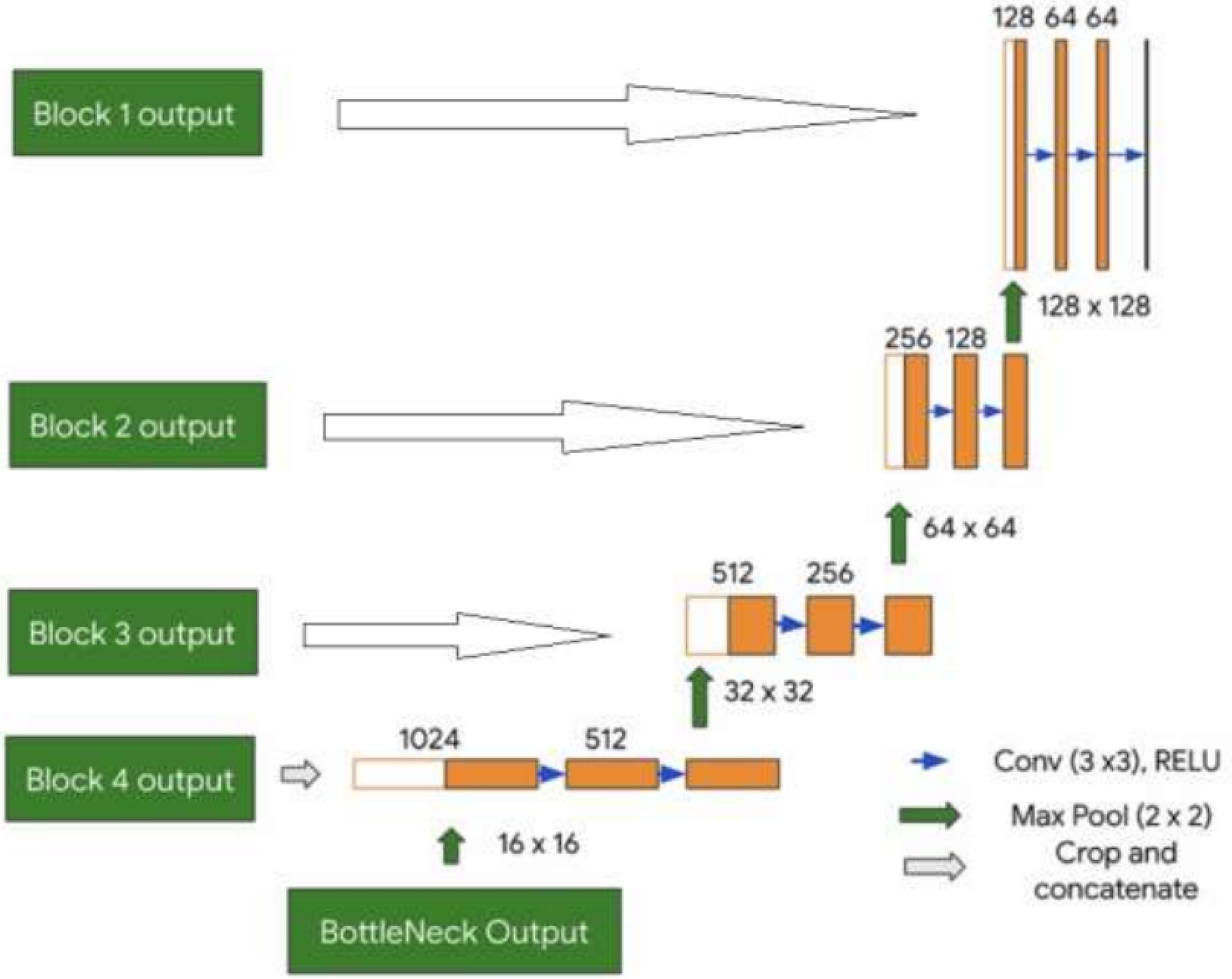
CNN u-net Structure



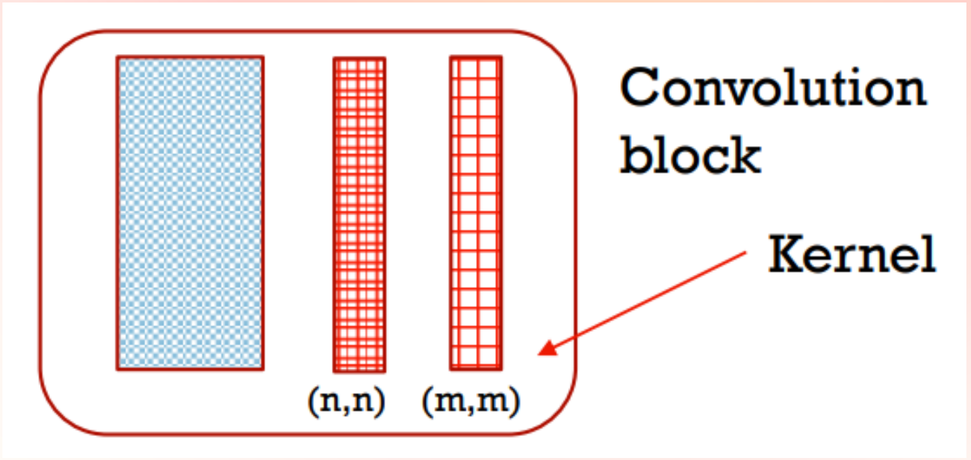
Down-sampling



Up-sampling

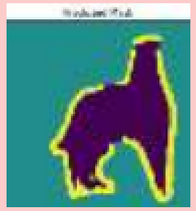

Block 1 output
128 64 64
128 x 128
Block 2 output
256 128
64 x 64
Block 3 output
512 256
32 x 32
Block 4 output
1024 512
16 x 16
BottleNeck Output
Conv (3 x3), RELU
Max Pool (2 x 2)
Crop and concatenate

Results



Input

Label

Architecture	Ouput image	Accuracy
Kernel – (3*3) n=m=3 dropout =30%		84.05%
Kernel – (3*3) n=m=3 No dropout layer		69.4%
Kernel – (5*5) n=m=5		63.87%

After adding Batch Normalization

background: IOU: 91.9441361524749

pet: IOU: 74.64327491311515

outline: IOU: 48.27249039024611

- Batch of 50 images choosen
- Batch normalization applies a transformation that maintains the mean output close to 0 and the output standard deviation close to 1.
- The layer will only normalize its inputs during inference after having been trained on data.
- Accuracy = 89%

Summary

- Selecting best hyperparameter is very important
- Batch normalization optimizes network training
- Dropping features in neural network solves the overfitting problem
- Skip connection helps to recover lost features while performing down sampling/max pooling
- Unet Architecture works very well in the segmentation domain

Video Compression

Now let's discuss about video compression

Steps Followed

- **Motion estimation and compression:** Used CNN model to estimate the optical flow. Then MV encoder decoder is used to compress and decode optical flow values.
- **Motion Compensation:** Obtain the predicted frame.
- **Transform, quantization and inverse transform:** Used non-linear encoder, decoder network, and quantization.
- **Entropy coding:** Quantized motion and residual representation are coded into bits and sent to the decoder.
- **Frame Reconstruction**

Encoder decoder and Motion compensation

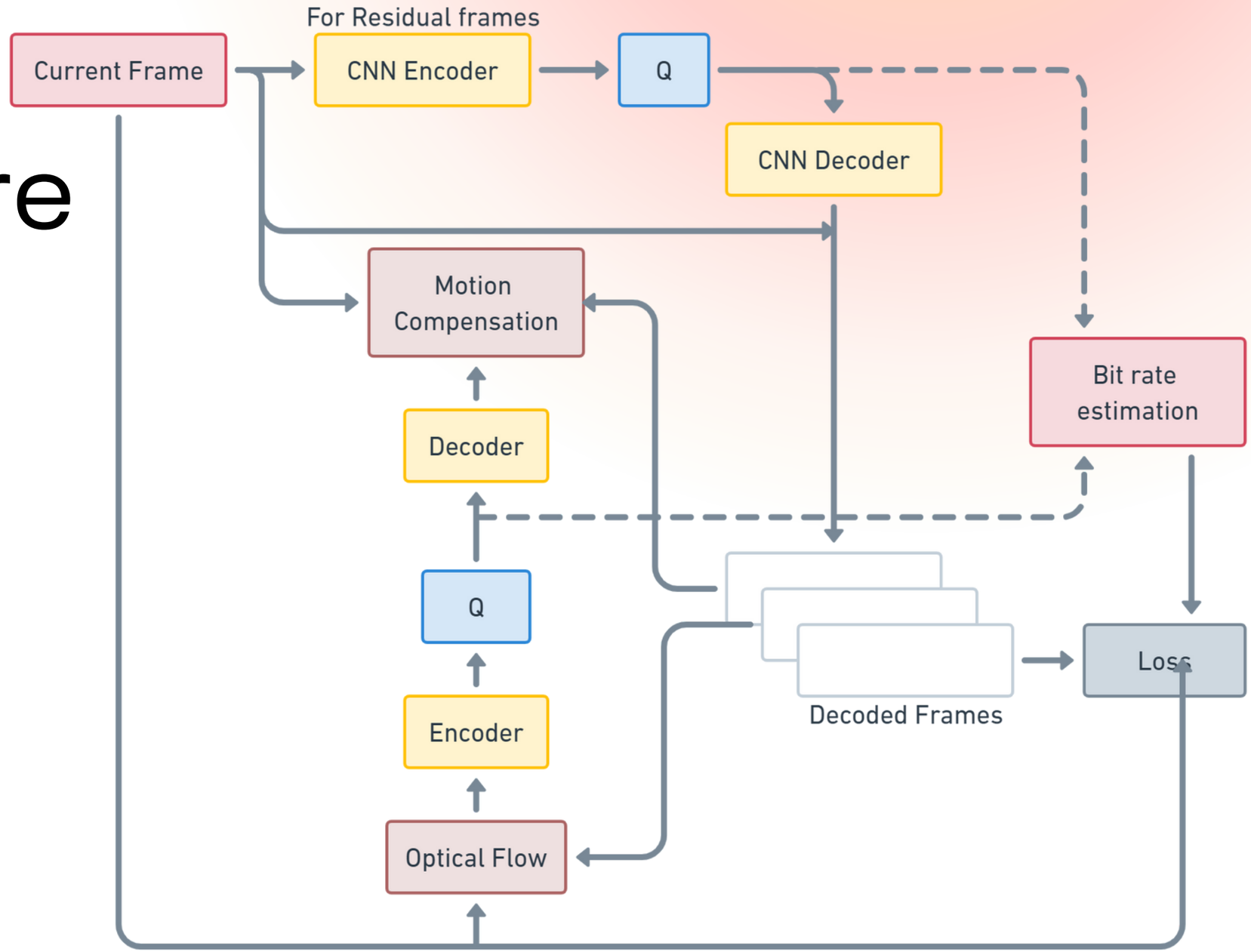
- We use CNN to transform the frames to corresponding representations.
- Optical flow is passed through series of convolutional layers and non linear transform.
- Encoder generates the motion representations.
- Decoder receives reconstruct information and quantized representation.
- CNN consist of Conv (3,128,2) kernel size 3x3 , output channel 128 and stride 2. And similarly Deconv (3, 128, 2) and last Deconv layer of (3,2,2).
- Previous frame is warped to current frame.
- To remove artifacts we concatenate warped frame, reference frame and motion vector.

Training and Results

- **Loss function:** Goal is to minimize the number of bits as well as distortion. We use Lagrange multiplier lambda that determines the tradeoff between distortion and number of bits.
- We also introduce uniform noise to address the quantization problem.
- **Bit Rate estimation:** Use entropy as our measure. Probability distributions are calculated from CNNs.
- Results include the variation of BPP with MS-SSIM parameter. Gain around 0.6 at same BPP level.

Architecture

Here is the overall architecture



Contributions

Here are the individual contributions of group members.

Members	Video Segmentation	Video Compression
Mayank Singh Rajput	Data analysis & visualization	Model Building and Testing
Mohit Ahirwar	U-Net architecture and training	Data analysis & visualization
Ram Khandelwaal	Model Building and Testing	Auto Encoder architecture and Training

Apart from these, there have been equal contribution in researching and presenting.

References

Video Segmentation

- https://arxiv.org/pdf/2107.01153.pdf : Used for survey , dataset and initial learning
- https://arxiv.org/abs/2209.01355 : Used for Encoder -decoder referance
- https://ieeexplore.ieee.org/document/9743897 : Used for data, analysis and strategy/

Video Compression

- https://www.arxiv-vanity.com/papers/2011.03029/ : Used for Compress AI
- https://arxiv.org/abs/2011.03029 : Used for initial learning and strategy
- https://arxiv.org/pdf/1904.03567.pdf : Used for Auto encoder and other implementation details

Github Link

- ramkhandelwal/Video-Processing (github.com)

Thanks

A ppt by Mayank, Mohit and Ram