SURGICAL TOOL ANALYSIS

V L Sahithi
IMT2020047

PROCESS

Taking video and text file as input
extracting frames
we have to use these frames to train the cnn we use
data augmentation
model architecture
optimizer
testing the model with testing data

FRAMES

Here we extract frames from video file and create a labeled dataset for tool presence detection task.
First we create a list of all the video files in the folder.
We define a dictonary to map the video file names to labels.
Now it loops through all the video files and extracts the frames.
Here we check if the vidoe file is in the babel dictonary and set tha label and load the text file for ths video which contains the instrument information.
Now we read the video file and check if it opened successfully or not and then we read the video frams and resize it to 224x224 pixels.
Now we create a one-hot encoded vector for the label and the instrument information and we append this to the list of lables.
Even the instrument info is converted to one-hot encoded array

DATA AUGMENTATION

Data augmentation refers to the process of artificially increasing the size of a training dataset by creating modified versions of the original images through a series of transformations, such as rotating, flipping, cropping, zooming, or changing the brightness or contrast of the images.

we initialize an instance of the ImageDataGenerator class from the Keras library. ImageDataGenerator is a powerful tool for data augmentation in deep learning, allowing for the generation of artificially augmented training data from a relatively small set of original images.

CNN MODEL-1

CNN MODEL 1:
first layer is a convolutional layer with 2 filters of size 3x3 and ReLU activation function. The input shape of the layer is (224, 224, 3),
The second layer is a Flatten layer, which converts the 2D feature maps generated by the convolutional layer into a 1D feature vector.
third layer is a fully connected layer with 9 neurons and a softmax activation function.

Than we compile the model using ADAM optimizer with learning rate = 0.0001
Loss function = categorical cross entropy

CNN MODEL-2

CNN MODEL 2:
first layer is Conv2D layer with 16 filters, each of size 3x3, using 'relu' activation function, and input_shape of (224, 224, 3).
MaxPooling2D layer with pool size of 2x2. This layer performs down-sampling by taking the maximum value in each 2x2 pool
Conv2D layer with 32 filters, each of size 3x3, using 'relu' activation function. This layer performs another round of convolution operation to extract more complex features.
MaxPooling2D layer with pool size of 2x2. than a Flatten layer.
Dense layer with 64 neurons using 'relu' activation function. This layer is a fully connected layer that performs a linear transformation
Dense layer with 1 neuron using 'sigmoid' activation function.

Than we compile the model using ADAM optimizer with learning rate = 0.0001
Loss function = categorical cross entropy

CNN MODELS

LE-NET - Test accuracy: 74.345
ZF-NET - Test accuracy: 62.347711672989625
GOOGLE-NET - Test accuracy: 59.592
ALEXNET - Test accuracy: 54.31580447797197
VGG - Test accuracy: 50.0
Polynet - 46.1289037095634

LENET

```
Accuracy for instrument class 1: 0.67
Accuracy for instrument class 2: nan
Accuracy for instrument class 3: 0.56
Accuracy for instrument class 4: nan
Accuracy for instrument class 5: nan
Accuracy for instrument class 6: nan
Accuracy for instrument class 7: nan
```

ZFNET

```
Accuracy for instrument class 1: 0.75
Accuracy for instrument class 2: nan
Accuracy for instrument class 3: 0.42
Accuracy for instrument class 4: nan
Accuracy for instrument class 5: nan
Accuracy for instrument class 6: nan
Accuracy for instrument class 7: nan
```

GOOGLE NET

```
Accuracy for instrument class 1: 0.53
Accuracy for instrument class 2: nan
Accuracy for instrument class 3: 0.34
Accuracy for instrument class 4: nan
Accuracy for instrument class 5: nan
Accuracy for instrument class 6: nan
Accuracy for instrument class 7: nan
```

Here we implemented a cnn for surgical tool detection in vidoes.

We built the model using keras framework

input to the model is 224x224x3 image

output is a one-hot encoded vector showing presence and absence of surgical tool and the type of tool detected in the video.

TRAINING CNN

We create data generator for training data and we use flow method that takes training data and generates batches of augmented data.

now we call fit on the model object with train_generator as the input and we train the model

TESTING

Now we have extract frames and resize it to 224x224 in the same way as we did for training. Text file is also loaded and we extract the instrument information here.

Next, a numpy array called **test_instruments_array** is initialized with zeros for each of the seven possible instruments.

Then, each instrument in the test_instruments list is checked and if it matches a certain instrument, the corresponding element in test_instruments_array is set to 1.

Finally, test_labels is created by concatenating a zero and the test_instruments_array for each frame in test_frames and then repeating this concatenated array for the total number of frames in test_frames. This will be used as the ground truth labels for the test video frames.

TESTING

After these frames are preprocessed we use the trained model to predict the label and instruments of each frame.

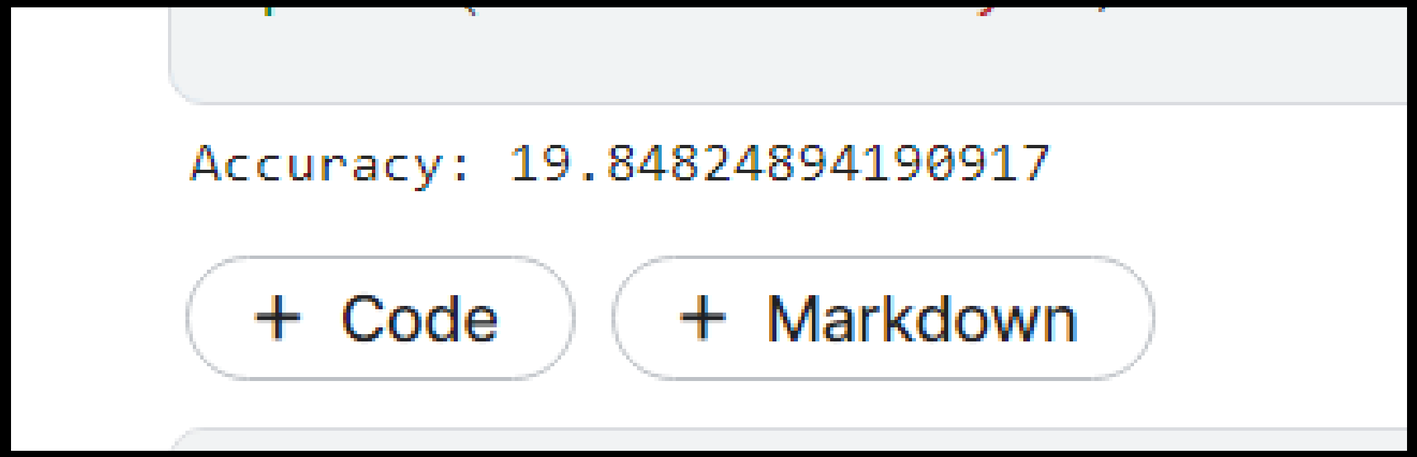
here the test_labels contains the target labels for the test_data the train_labels are used to train the model and than we evaluate it using test.

test_predictions are the predicted labels generated by the trained model for the test images. **model.predict(test_images)** predicts the class labels for the test images.

test_labels are the true labels of the test images.

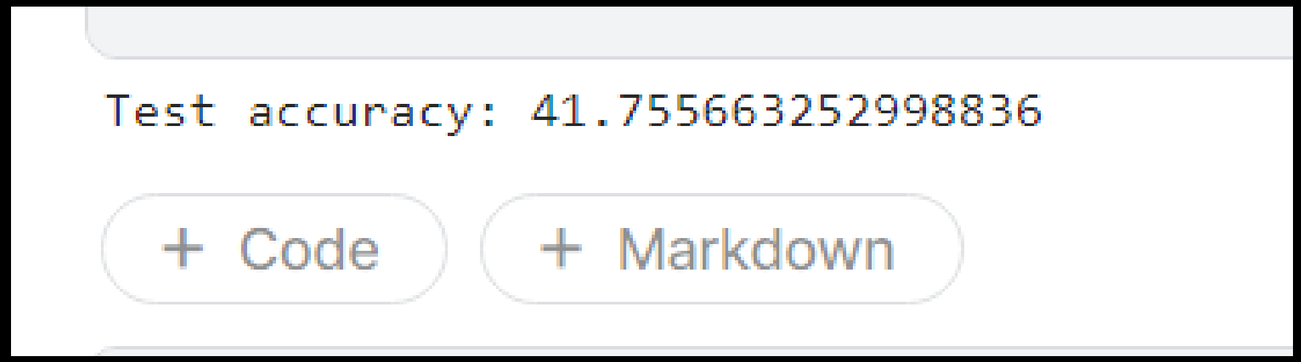
TESTING CNN-1

Now we calculate the test accuracy of a machine learning model by comparing its predicted outputs to the actual labels for a given test dataset.

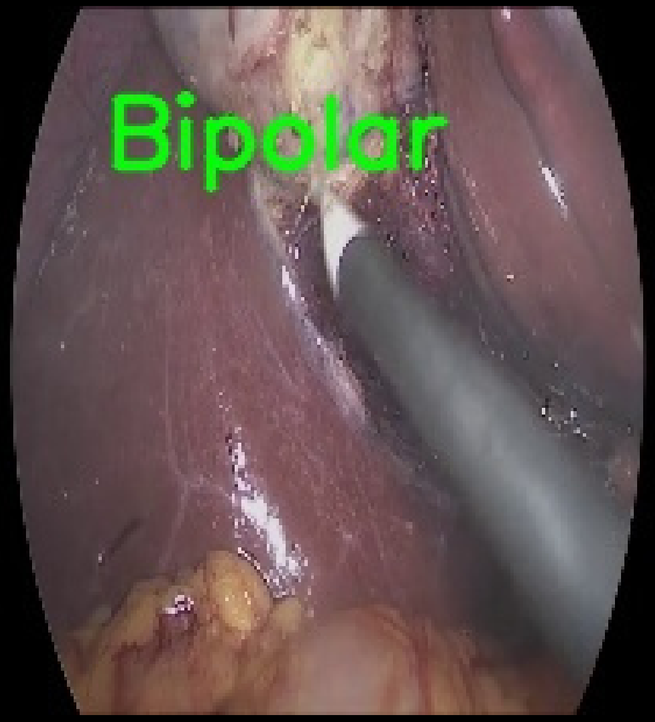


TESTING CNN-2

Now we calculate the test accuracy of a machine learning model by comparing its predicted outputs to the actual labels for a given test dataset.



CNN-1



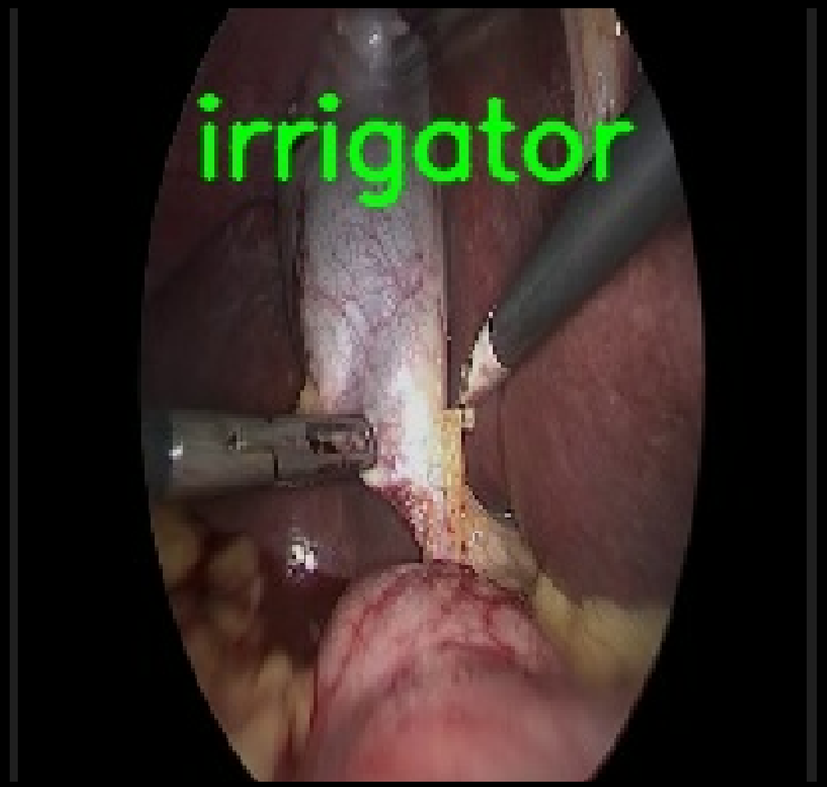
CNN-2

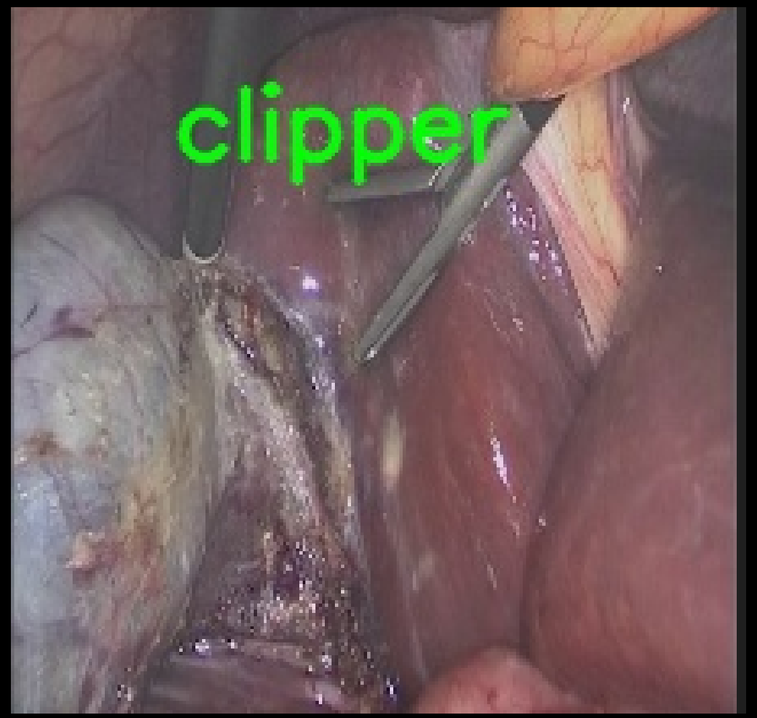
LE-NET

ZFNET

GOOGLENET

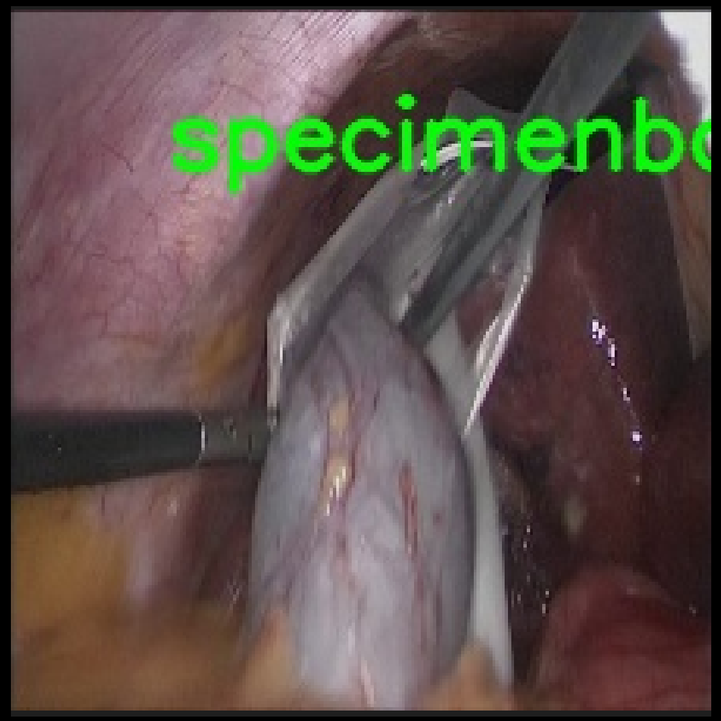
ALEXNET



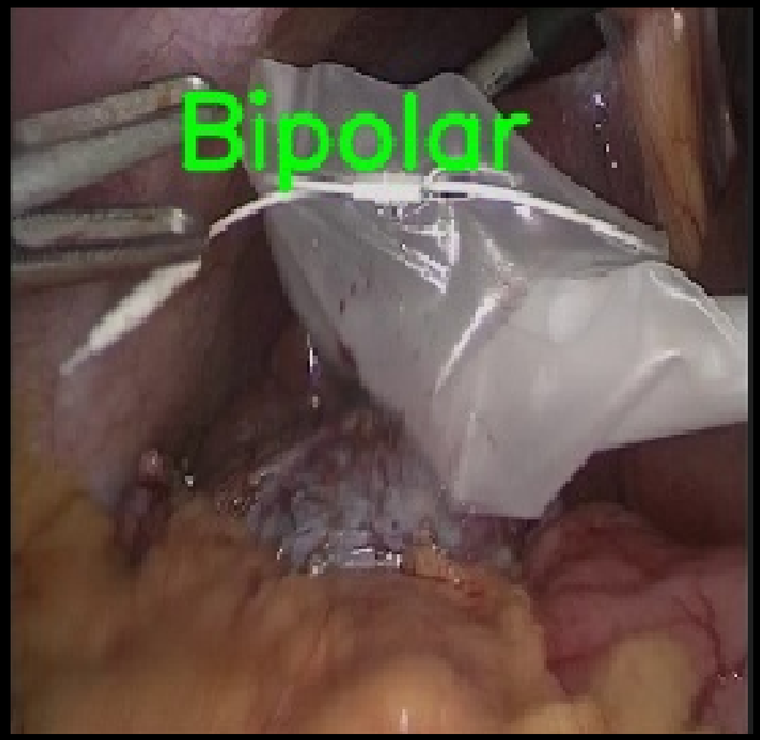

clipper


grasper


irrigator


specimenbo


Bipolar


Bipolar

RESAMPLING

Resampling is a technique used to deal with imbalanced datasets, where one class has significantly fewer samples than the other classes.

The resampling technique used here is oversampling, where the minority instrument class is sampled with replacement to create additional samples until it is balanced with the majority class. This helps to address the class imbalance issue and ensure that the model is not biased towards the majority class.

ZFNET

```
Test loss: 8.14467834
Test accuracy: 82.6934445
```

TRAINING WITH 3 VIDEOS AND USING TRAIN - TEST SPLIT

FUNCTIONS:
MERGE_DATA
PREPROCESS
CREATE_MODEL

LENET

```
Test loss: 3.528920888900757
Test accuracy: 94.45685958862
```

	precision	recall	f1-score	support
0	0.50	0.90	0.67	846
1	0.00	0.00	0.00	40
2	0.45	0.82	0.56	536
3	0.00	0.00	0.00	15
4	0.00	0.00	0.00	19
5	0.00	0.00	0.00	27
6	0.25	0.36	0.10	80