Date: June 11, 2020

Please, skim though this before you watch today's video.

Today, we are going to talk about basic matrix operations in OpenCV.

-Matrix Initialization.

There are several multiple ways to initialize your matrix in OpenCV

1.Initialization with Zero Matrix

You can initialize a Mat object that has 0s for all components.

e.g.

```
//1.Initialization with zero matrix
int rowSize = 3;
int colSize = 3;
Size matrixSize1(rowSize, colSize);
Mat zeroMatrix = Mat::zeros(matrixSize1, CV_32FC1);//3x3 zero matrix with 1 channel.

cout << zeroMatrix << endl;
```

```
[0, 0, 0;
 0, 0, 0;
 0, 0, 0]
```

2.Initialization with One Matrix

Similarly, you can define a Mat object with matrix that has 1s for all components

e.g.

```
//2.Initialization with one matrix
Mat oneMatrix = Mat::ones(matrixSize1, CV_32FC1);//3x3 one matrix with 1 channel.

cout << oneMatrix << endl;
```

```
[1, 1, 1;
 1, 1, 1;
 1, 1, 1]
```

Date: June 11, 2020

3.Initialization with Identity Matrix

One of the most important matrices in linear algebra is the identity matrix. We can define this as follows.

e.g.

```
//3.Initialization with identity matrix
Mat identityMatrix = Mat::eye(matrixSize1, CV_32FC1);//3x3 identity matrix with 1 channel.

cout << identityMatrix << endl;
```

```
[1, 0, 0;
 0, 1, 0;
 0, 0, 1]
```
f

-Basic Matrix Operations

With power of OpenCV, many operations you can possibly think of with matrices can be performed. Here, we are going to take a look at matrix addition, matrix subtraction, element-wise multiplication, and matrix multiplication. There are a lot more matrix operations. We will learn additional matrix operations as we need.

(1) Matrix Addition

We can perform element-wise addition with simple operator “+”.

e.g.

```
//(1) Matrix addition
Mat resultMat1 = identityMatrix + identityMatrix;
cout << "Matrix addition: " << endl;
cout << resultMat1 << endl;
```

```
[1, 0, 0;
 0, 1, 0;
 0, 0, 1]
```
+
```
[1, 0, 0;
 0, 1, 0;
 0, 0, 1]
```
=
```
[2, 0, 0;
 0, 2, 0;
 0, 0, 2]
```

(2) Matrix Subtraction

We can perform element-wise addition with a simple operator “-”.

e.g.

```
//(2) Matrix subtraction
Mat resultMat2 = identityMatrix - identityMatrix;
cout << "Matrix subtraction: " << endl;
cout << resultMat2 << endl;
```

```
[1, 0, 0;
 0, 1, 0;
 0, 0, 1]
```
-
```
[1, 0, 0;
 0, 1, 0;
 0, 0, 1]
```
=
```
[0, 0, 0;
 0, 0, 0;
 0, 0, 0]
```

(3) Element-wise multiplication

We can perform element-wise subtraction with ".mul()" function. Note that there is a dot before "mul()".

e.g.

```
//(3) Element-wise matrix multiplication
Mat resultMat3 = resultMat1.mul(resultMat1);
cout << "Element-wise matrix multiplication: " << endl;
cout << resultMat3 << endl;
```

```
[2, 0, 0;
 0, 2, 0;
 0, 0, 2]
```
.X
```
[2, 0, 0;
 0, 2, 0;
 0, 0, 2]
```
=
```
[4, 0, 0;
 0, 4, 0;
 0, 0, 4]
```

(4) Matrix Multiplication

We can perform matrix multiplication with simple operator "*" (asterisk).

```
//(4) Matrix multiplication
Mat tempMat1 = Mat::ones(4, 2, CV_32FC1);
Mat tempMat2 = Mat::ones(2, 4, CV_32FC1);

Mat resultMat4 = tempMat1*tempMat2;
```

e.g.

Date: June 11, 2020