

영상처리 실제 - 14주차 실습

: 19.영상분류 - p.9 ~ 10

```
//19 - 영상분류 - p.9 ~ 10
if 1
    Mat train_features(5, 2, CV_32FC1);
    Mat labels(5, 1, CV_32FC1);

    // 점의 좌표를 train_features에 입력한다.
    train_features.at<float>(0, 0) = 10, train_features.at<float>(0, 1) = 10;
    train_features.at<float>(1, 0) = 10, train_features.at<float>(1, 1) = 20;
    train_features.at<float>(2, 0) = 20, train_features.at<float>(2, 1) = 10;
    train_features.at<float>(3, 0) = 30, train_features.at<float>(3, 1) = 30;
    train_features.at<float>(4, 0) = 40, train_features.at<float>(4, 1) = 30;

    // 원하는 레이블을 labels에 입력한다.
    labels.at<float>(0, 0) = 1;
    labels.at<float>(1, 0) = 1;
    labels.at<float>(2, 0) = 1;
    labels.at<float>(3, 0) = 2;
    labels.at<float>(4, 0) = 2;

    // 학습 과정
    Ptr<ml::KNearest> knn = ml::KNearest::create();
    Ptr<ml::TrainData> trainData = ml::TrainData::create(train_features, ml::ROW_SAMPLE, labels);
    knn->train(trainData);

    // 테스트 과정
    Mat sample(1, 2, CV_32FC1);
    Mat predictedLabels;

    // 테스트 데이터를 입력한다.
    sample.at<float>(0, 0) = 28, sample.at<float>(0, 1) = 28;
    knn->findNearest(sample, 2, predictedLabels);

    float prediction = predictedLabels.at<float>(0, 0);
    cout << "테스트 샘플의 라벨 = " << prediction << endl;
    waitKey();
endif
```

Microsoft Visual Studio 디버그 콘솔

```
테스트 샘플의 라벨 = 2
[ INFO:0@0.132] global c:\build\master_winpack-build-win64-vc14\opencv\modules\highgui\src\registry.impl.hpp (114) cv::highgui_backend::UIBackendRegistry::UIBackendRegistry UI: Enabled backends(4, sorted by priority): GTK(1000); GTK3(990); GTK2(980); WIN32(970) + BUILTIN(WIN32UI)
[ INFO:0@0.132] global c:\build\master_winpack-build-win64-vc14\opencv\modules\core\src\utils\plugin_loader.impl.hpp (67) cv::plugin::impl::DynamicLib::libraryLoad load D:\#1.???#2.???????#2.23?2??(??2??)#2.?????#3.??#1.SRC\Week_Practice#x64\Debug\opencv_highgui_gtk460_64.dll => FAILED
[ INFO:0@0.133] global c:\build\master_winpack-build-win64-vc14\opencv\modules\core\src\utils\plugin_loader.impl.hpp (67) cv::plugin::impl::DynamicLib::libraryLoad load opencv_highgui_gtk460_64.dll => FAILED
[ INFO:0@0.133] global c:\build\master_winpack-build-win64-vc14\opencv\modules\core\src\utils\plugin_loader.impl.hpp (67) cv::plugin::impl::DynamicLib::libraryLoad load D:\#1.???#2.???????#2.23?2??(??2??)#2.?????#3.??#1.SRC\Week_Practice#x64\Debug\opencv_highgui_gtk3460_64.dll => FAILED
[ INFO:0@0.134] global c:\build\master_winpack-build-win64-vc14\opencv\modules\core\src\utils\plugin_loader.impl.hpp (67) cv::plugin::impl::DynamicLib::libraryLoad load opencv_highgui_gtk3460_64.dll => FAILED
[ INFO:0@0.135] global c:\build\master_winpack-build-win64-vc14\opencv\modules\core\src\utils\plugin_loader.impl.hpp (67) cv::plugin::impl::DynamicLib::libraryLoad load D:\#1.???#2.???????#2.23?2??(??2??)#2.?????#3.??#1.SRC\Week_Practice#x64\Debug\opencv_highgui_gtk2460_64.dll => FAILED
[ INFO:0@0.135] global c:\build\master_winpack-build-win64-vc14\opencv\modules\core\src\utils\plugin_loader.impl.hpp (67) cv::plugin::impl::DynamicLib::libraryLoad load opencv_highgui_gtk2460_64.dll => FAILED
[ INFO:0@0.136] global C:\build\master_winpack-build-win64-vc14\opencv\modules\highgui\src\backend.cpp (90) cv::highgui_backend::createUIBackend UI: using backend: WIN32 (priority=970)

D:\#1.개인폴더#2.산업인공지능학과#2.23년2학기(석사2학기)#2.영상처리실제#3.실습#1.SRC\Week_Practice#x64\Debug\Week_Practice.exe(프로세스 4612개)이(가) 종료되었습니다(코드: 0개).
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [디버깅] > [디버깅이 중지되면 자동으로 콘솔 닫기]를 사용하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

: 19.영상분류 - p.12 ~ 13

```
//19 - 영상분류 - p.12 ~ 13
#include 1
Mat img;
img = imread("D:\\999.Image\\digits.png", IMREAD_GRAYSCALE);
namedWindow("original", WINDOW_AUTOSIZE);
imshow("original", img);
waitKey(0);

Mat train_features(5000, 400, CV_32FC1);
Mat labels(5000, 1, CV_32FC1);

// 각 숫자 영상을 행 벡터로 만들어서 train_feature에 저장한다.
for (int r = 0; r < 50; r++)
{
    for (int c = 0; c < 100; c++)
    {
        int i = 0;
        for (int y = 0; y < 20; y++)
        {
            for (int x = 0; x < 20; x++)
            {
                train_features.at<float>(r * 100 + c, i++) = img.at<uchar>(r * 20 + y, c * 20 + x);
            }
        }
    }
}

// 각 숫자 영상에 대한 레이블을 저장한다.
for (int i = 0; i < 5000; i++)
{
    labels.at<float>(i, 0) = (i / 500);
}

// 학습 과정
Ptr<ml::KNearest> knn = ml::KNearest::create();
Ptr<ml::TrainData> trainData = ml::TrainData::create(train_features, ml::ROW_SAMPLE, labels);
knn->train(trainData);

// 테스트 과정
Mat predictedLabels;
for (int i = 0; i < 5000; i++)
{
    Mat test = train_features.row(i);
    knn->findNearest(test, 3, predictedLabels);
    float prediction = predictedLabels.at<float>(0);
    cout << "테스트 샘플" << i << "의 라벨 = " << prediction << '\n';
}
waitKey();
#endif
```

```
D:\#1.개인폴더\#2.산업인공지능학과\#2.23년2학기(석사2학기)\#2.영상처리실제\#3.실습\#1.SRC\Week_Practice\#x64\Debug\Week_Practice.exe
테스트 샘플 4971의 라벨 = 9
테스트 샘플 4972의 라벨 = 9
테스트 샘플 4973의 라벨 = 9
테스트 샘플 4974의 라벨 = 9
테스트 샘플 4975의 라벨 = 9
테스트 샘플 4976의 라벨 = 9
테스트 샘플 4977의 라벨 = 9
테스트 샘플 4978의 라벨 = 9
테스트 샘플 4979의 라벨 = 9
테스트 샘플 4980의 라벨 = 9
테스트 샘플 4981의 라벨 = 9
테스트 샘플 4982의 라벨 = 9
테스트 샘플 4983의 라벨 = 9
테스트 샘플 4984의 라벨 = 9
테스트 샘플 4985의 라벨 = 9
테스트 샘플 4986의 라벨 = 9
테스트 샘플 4987의 라벨 = 9
테스트 샘플 4988의 라벨 = 9
테스트 샘플 4989의 라벨 = 9
테스트 샘플 4990의 라벨 = 9
테스트 샘플 4991의 라벨 = 9
테스트 샘플 4992의 라벨 = 9
테스트 샘플 4993의 라벨 = 9
테스트 샘플 4994의 라벨 = 9
테스트 샘플 4995의 라벨 = 9
테스트 샘플 4996의 라벨 = 9
테스트 샘플 4997의 라벨 = 9
테스트 샘플 4998의 라벨 = 9
테스트 샘플 4999의 라벨 = 9
```

: 19.영상분류 – p.19 ~ 20

```
//19 - 영상분류 - p.19 ~ 20
#ifdef OPENCV
#include <opencv2/core.hpp>
#include <opencv2/imgproc.hpp>
#include <opencv2/highgui.hpp>
using namespace cv;

int main()
{
    if 1
    {
        Mat samples(50, 2, CV_32F);

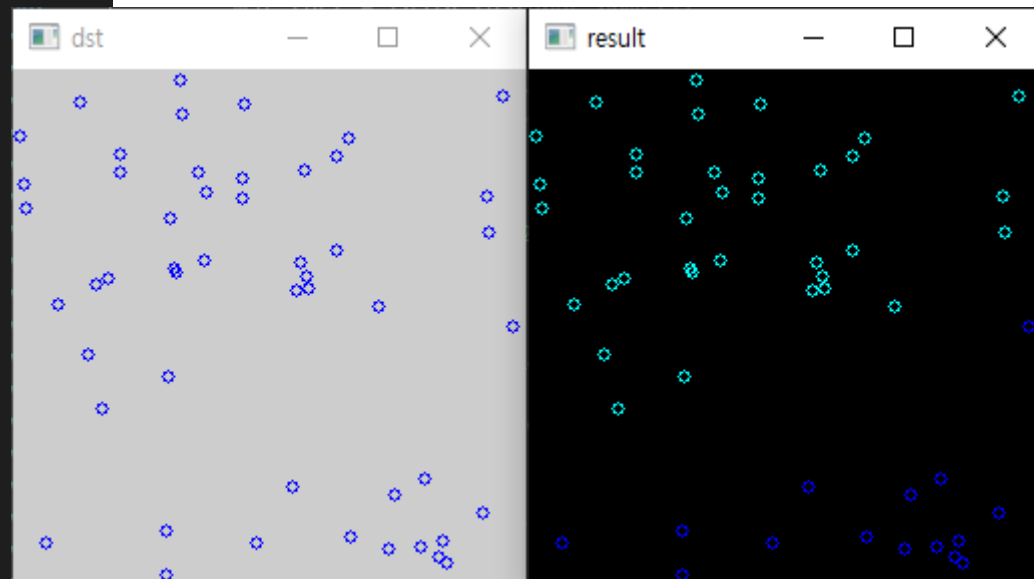
        for (int y = 0; y < samples.rows; y++)
        {
            samples.at<float>(y, 0) = (rand() % 255);
            samples.at<float>(y, 1) = (rand() % 255);
        }
        Mat dst(256, 256, CV_8UC3);

        for (int y = 0; y < samples.rows; y++)
        {
            float x1 = samples.at<float>(y, 0);
            float x2 = samples.at<float>(y, 1);
            circle(dst, Point(x1, x2), 3, Scalar(255, 0, 0));
        }
        imshow("dst", dst);

        Mat result;
        Mat labels(50, 1, CV_8UC1);

        Mat centers;
        result = Mat::zeros(Size(256, 256), CV_8UC3);
        kmeans(samples, 2, labels, TermCriteria(TermCriteria::MAX_ITER | TermCriteria::EPS, 10000, 0.0001),
            3, KMEANS_PP_CENTERS, centers);

        for (int y = 0; y < samples.rows; y++)
        {
            float x1 = samples.at<float>(y, 0);
            float x2 = samples.at<float>(y, 1);
            int cluster_idx = labels.at<int>(y, 0);
            if (cluster_idx == 0)
                circle(result, Point(x1, x2), 3, Scalar(255, 0, 0));
            else
                circle(result, Point(x1, x2), 3, Scalar(255, 255, 0));
        }
        imshow("result", result);
        waitKey(0);
    }
#endif
}
```



: 19.영상분류 – p.21

```
//19 - 영상분류 - p.21
if 1
    Mat src = imread("D:\\999.Image\\lenna.jpg", 1);

    // 학습 데이터를 만든다.
    Mat samples(src.rows * src.cols, 3, CV_32F);
    for (int y = 0; y < src.rows; y++)
    {
        for (int x = 0; x < src.cols; x++)
        {
            for (int z = 0; z < 3; z++)
            {
                samples.at<float>(y + x * src.rows, z) = src.at<Vec3b>(y, x)[z];
            }
        }
    }

    // 클러스터의 개수는 15가 된다.
    int clusterCount = 15;
    Mat labels;
    int attempts = 5;
    Mat centers;
    kmeans(samples, clusterCount, labels, TermCriteria(TermCriteria::MAX_ITER | TermCriteria::EPS, 10000, 0.0001),
        attempts, KMEANS_PP_CENTERS, centers);

    Mat new_image(src.size(), src.type());
    for (int y = 0; y < src.rows; y++)
    {
        for (int x = 0; x < src.cols; x++)
        {
            int cluster_idx = labels.at<int>(y + x * src.rows, 0);
            new_image.at<Vec3b>(y, x)[0] = centers.at<float>(cluster_idx, 0);
            new_image.at<Vec3b>(y, x)[1] = centers.at<float>(cluster_idx, 1);
            new_image.at<Vec3b>(y, x)[2] = centers.at<float>(cluster_idx, 2);
        }
    }
    imshow("clustered image", new_image);
    waitKey(0);
endif
```

