در ابتدا با استفاده از کراس دیتا اموزشی را augment می کنیم

```
from keras.preprocessing.image import ImageDataGenerator

# Define the image data generator with augmentation techniques
train_datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.3,
    horizontal_flip=True,
    fill_mode='nearest')
```

چون داده های با label صفر ۶۴ درصد داده ها را تشکیل می دهند پس در داده های ما توازن ندارد برای برقراری توازن از SMOTE استفاده می کنیم یک تعریفی از SMOTE که دقیقا چی هست:

وقتی چنین مشکلی داریم یعنی دیتاست ما نامتعادل هست ما معمولا چهارتا راه داریم:

1. Weighted Class Approach
2. Under-sampling approach
3. Data Augmentation for Minority Class
4. Synthetic Minority Over-sampling Technique (SMOTE)

در تکنیک هایی که دربالا اشاره شد data augmentation,SMOTE از بقیه بهتر هستند که استفاده از SMOTE رایج تر هست.و برای متعادل سازی دیتاست نیز از SMOTE استفاده کردم.

در ادامه callback های مربوط به مدل را تعریف می کنیم و بعد داده های اموزشی را ۰.۲ ان را به داده های validation اختصاص میدم و در ادامه ابتدا یک ورودی تصویری با استفاده از کلاس ImageInput از کتابخانه AutoKeras تعریف می‌شود. سپس با استفاده از کلاس Normalization یک لایه نرمال‌سازی اعمال می‌شود سپس با استفاده از کلاس ImageAugmentation، یک لایه Augmentation به صورت افقی روی تصاویر اعمال می‌شود. این کار به شبکه عصبی کمک می‌کند تا بتواند تصاویر را با تنوع بیشتری یاد بگیرد.سپس با استفاده

از کلاس ResNetBlock، یک بلوک از مدل ResNet ساخته می‌شود. این بلوک شامل چندین لایه پیچشی و ادغام است و در شبکه عصبی به کمک انتقال رو به جلو، اطلاعات را از ورودی به خروجی منتقل می‌کند.در نهایت، با استفاده از کلاس ClassificationHead، یک لایه خروجی برای دسته‌بندی تصاویر تعریف می‌شود. این لایه شامل یک لایه GlobalAveragePooling و یک لایه Dense است.در بخش آخر، با استفاده از کلاس AutoModel از کتابخانه AutoKeras، یک مدل شبکه عصبی پیچشی به صورت خودکار با استفاده از چندین مدل مختلف، طراحی می‌شود. با تعیین مقادیر پارامترهای مختلف مانند تعداد تلاش‌های انجام شده برای طراحی مدل و هدف بهینه‌سازی، این کد به AutoKeras امکان می‌دهد تا یک مدل عصبی مناسب برای دسته‌بندی تصاویر را به صورت خودکار ساخته و بر روی داده‌های آموزشی، آموزش دهد. و بعد از train کردن مدل نتایج زیر به دست امد:

بر روی داده های تست دقت به صورت زیر است:

```
(749, 100, 100, 3)
24/24 [==============================] - 6s 151ms/step
Test accuracy: 0.8958611481975968
```

و بر روی داده های train دقت به صورت زیر است:

```
(749, 100, 100, 3)
24/24 [==============================] - 6s 151ms/step
Test accuracy: 0.8958611481975968
```

نتایج بالا بر روی colab چند روز گذاشتیم تا ران بشود و بر روی GPU ازمایشگاه نیز بیش از دو روز گذاشتم تا بر روی شبکه های عصبی سرچ انجام شود و نتایج به صورت زیر شد برای داده های تست به دقت حدود ۹۱ درصد رسیدیم:

```
24/24 [==============================] - 30s 1s/step
Test accuracy: 0.9092122830440588
```

و برای داده های اموزشی به دقت حدود ۹۴ درصد رسیدیم:

```
91/91 [==============================] - 114s 1s/step
Test accuracy: 0.9467680608365019
```