

پروژه: ساخت بازی سنگ، کاغذ، قیچی با هوش مصنوعی

بخش اول: تعریف دقیق مسئله

بازی سنگ، کاغذ، قیچی رو که می‌شناسی؟ یه بازی ساده و قدیمی که دو نفر هم‌زمان یکی از این سه حالت رو انتخاب می‌کنن و برنده بر اساس این قانون‌ها مشخص می‌شه:

- سنگ قیچی رو می‌بره ✂️👊
- کاغذ سنگ رو می‌بره 🤲👊
- قیچی کاغذ رو می‌بره 🤲✂️

حالا قراره یه نسخه پیشرفته‌تر از این بازی بسازیم که به جای دوتا بازیکن واقعی، یکی از اون‌ها **هوش مصنوعی (AI)** باشه! یعنی ما جلوی دوربین یکی از این حرکت‌ها رو نشون می‌دیم و کامپیوتر تشخیص می‌ده که سنگ، کاغذ یا قیچی هست، بعدش یه حرکت تصادفی انجام می‌ده و نتیجه بازی رو مشخص می‌کنه.

بخش دوم: آموزش مدل با Machine Teachable

برای اینکه هوش مصنوعی بتونه تشخیص بده که دست ما چه شکلیه، باید اول یه مدل یادگیری ماشینی بسازیم و بهش یاد بدیم که سنگ، کاغذ و قیچی چه شکلی هستن. این کار رو با ابزار **Machine Teachable** انجام می‌دیم که خیلی راحت و بدون نیاز به برنامه‌نویسی کار می‌کنه.

۱. جمع‌آوری داده‌ها

برای آموزش مدل، باید بهش کلی نمونه نشون بدیم.

- دوربین وبکم رو روشن می‌کنیم و دستمون رو توی سه حالت مختلف (سنگ، کاغذ، قیچی) جلوی دوربین می‌گیریم.
- از هر حرکت چندین نمونه ضبط می‌کنیم تا مدل بتونه اون‌ها رو بهتر یاد بگیره.
- بهتره نور محیط مناسب باشه و زوایای مختلف رو هم امتحان کنیم تا دقت مدل بالاتر بره.

۲. تعریف کلاس‌ها

ما سه کلاس مختلف داریم:

- کلاس سنگ 🖐️ (دستی که به شکل مشت بسته است)
- کلاس کاغذ 🖐️ (دستی که کاملاً باز است)
- کلاس قیچی ✂️ (دستی که دو انگشت باز دارد)

وقتی داده‌ها رو ضبط کردیم، می‌تونیم روی "Train Model" کلیک کنیم تا مدل شروع به یادگیری کنه.

۳. تست و بهبود مدل

بعد از آموزش، حالا باید ببینیم که مدل چقدر خوب یاد گرفته!

- دستمون رو جلوی دوربین می‌گیریم و مدل باید تشخیص بده که سنگ، کاغذ یا قیچی هست.
- اگه زیاد اشتباه می‌کنه، می‌تونیم نمونه‌های بیشتری بهش بدیم یا مدل رو دوباره آموزش بدیم.

۴. دانلود مدل

بعد از اینکه مدل به خوبی یاد گرفت، می‌تونیم اون رو **دانلود** کنیم. این مدل به صورت یک فایل با نام **keras_model.h5** ذخیره می‌شه که بعداً توی برنامه پایتونی ازش استفاده می‌کنیم.

بخش سوم: به‌کارگیری مدل در برنامه پایتونی

حالا وقتشه که مدلی که ساختیم رو توی یه برنامه واقعی استفاده کنیم! توی این مرحله، ما یک کد پایتونی می‌نویسیم که:

1. مدل یادگیری ماشین رو لود کنه.
2. دوربین رو روشن کنه و تصویر دست رو دریافت کنه.
3. پیش‌بینی کنه که حرکت دست چی هست (سنگ، کاغذ یا قیچی).
4. خودش یه حرکت تصادفی انتخاب کنه.
5. نتیجه بازی رو مشخص کنه.

۱. لود کردن مدل

مدلی که دانلود کردم رو داخل یک پوشه قرار میدیم، همه اینهارو هم داخل پوشه اصلی میذاریم که فایلها با هم قاطی پاتی نشن، بعد مدلمون رو لود میکنیم:

```
from tensorflow.keras.models import load_model
import cv2
import numpy as np

model = load_model("keras_model.h5")
print("✅ مدل با موفقیت بارگذاری شد")
```

۲. پردازش تصویر از دوربین

حالا باید دوربین رو باز کنیم و تصویر دستمون رو دریافت کنیم:

```
cap = cv2.VideoCapture(0)
if not cap.isOpened():
    print("خطا در باز کردن دوربین")
```

۳. پیش‌بینی مدل

وقتی دوربین روشن شد، تصویر رو می‌گیریم و به مدل می‌دیم تا تشخیص بده، مدل اندازه تصویر رو تغییر میده، یک سری چیزا بهش اضافه میکنه و آمادهش میکنه تا بتونه تشخیصش بده، در نهایت از دوربین ما قاره یک تصویر یا "فریم" بگیره و بفهمه سنگه، کاغذه، یا قیچی؟

```
ret, frame = cap.read()
if ret:
    img = cv2.resize(frame, (224, 224)) # تغییر اندازه تصویر به
    # اندازه ورودی مدل
    img = np.expand_dims(img, axis=0) # اضافه کردن بعد اضافی
    # برای مدل
    img = img / 255.0 # نرمال‌سازی داده‌ها

    predictions = model.predict(img)
    move = np.argmax(predictions)
    moves = ["سنگ", "کاغذ", "قیچی"]
    player_move = moves[move]
    print(f"!انتخاب کردید {player_move} شما 🖐️")
```

۴. انتخاب حرکت تصادفی کامپیوتر و اعلام نتیجه

حالا که حرکت بازیکن مشخص شد، باید کامپیوتر هم به حرکت تصادفی انتخاب کنه و نتیجه بازی رو حساب کنیم

برنامه پایتونی ما یک منطق ساده داره، اگه حرکت من و کامپیوتر یکی بود، مساوی شدیم!

اگر من سنگ انتخاب کردم و کامپیوتر کاغذ، یا من کاغذ انتخاب کردم و کامپیوتر سنگ یا من قیچی انتخاب کردم و کامپیوتر کاغذ، من بازی رو بردم! سه تا حرکت هست که ما میبریم، سه تا حرکت (هردوتا سنگ یا کاغذ یا قیچی) هم هست که مساوی میشه، من گفتم اگه این اتفاقا افتاد چی میشه، پس هر حالتی بجز اینا همیشه باخت ما:)

نوشتنش داخل پایتون به این صورت میشه:

```
import random

computer_move = random.choice(["سنگ", "کاغذ", "قیچی"])
print(f"!انتخاب کرد {computer_move} کامپیوتر")

if player_move == computer_move:
    print("!مساوی شد")
elif (player_move == "سنگ" and computer_move == "قیچی") or \
     (player_move == "کاغذ" and computer_move == "سنگ") or \
     (player_move == "قیچی" and computer_move == "کاغذ"):
    print("!شما برنده شدید")
else:
    print("!شما باختید")
```

۵. بستن دوربین

در آخر، باید دوربین رو ببندیم چون دیگه باهاش کاری نداریم

```
cap.release()  
cv2.destroyAllWindows()
```

نتیجه نهایی

حالا شما یه بازی هوشمند "سنگ، کاغذ، قیچی" دارید که حرکت دست شما رو تشخیص
میده و با شما بازی می‌کنه!  