

# برنامج محاثة فيديو باستخدام المقابس

ليال محرز، مازن شيبان

## ملخص:

برمجة المقابس هي طريقة لربط عقدتين على شبكة للتواصل مع بعضهما البعض. الفيديو عبارة عن سلسلة من الصور يتم عرضها بسرعة كبيرة بحيث يمكن للعقل البشري أن يعتقد أنها مشاهد متحركة باستمرار. الصورة مرة أخرى عبارة عن مجموعة من وحدات البكسل يتم تخزينها كقائمة من القوائم في مصفوفة عددية. لذلك لإنشاء تطبيق دردشة فيديو، يتعين علينا النقاط الصور ونقلها بسرعة عالية جداً. يتم تنفيذ هاتين المهمتين بواسطة مكتبات OpenCV و Socket على التوالي. لإدارة الصور ومعالجتها، نستخدم وحدات numpy و OpenCV. نستخدم وحدة threading لمتابعة إرسال واستقبال الصور بشكل متوازي. في هذه المقالة سنقوم بإنشاء تطبيق محاثة عبر الفيديو بين شخصين وذلك بالاستفادة من لغة بايثون ومكتباتها الشهيرة المذكورة سابقاً.

الكلمات المفتاحية: فيديو، بروتوكول TCP، open cv

# Video conference software using sockets

Layal mahrez, Mazen Shibani

## **Abstract:**

Socket programming is a method of connecting two nodes on a network to communicate with each other. A video is a series of images that are displayed so quickly that the human brain can think of them as constantly moving scenes. The image is again an array of pixels stored as a list of lists in an integer array. So to create a video chat app, we have to take pictures and transfer them at a very high speed. These two tasks are performed by the OpenCV and Socket libraries respectively. For image management and processing, we use numpy and OpenCV modules. The threading module is used to continue sending and receiving images in parallel. In this article, we are going to create a one-to-one video chatting app by taking advantage of the popular Python language and its previously mentioned libraries.

**Keywords:** video, TCP protocol, open cv

## مقدمة:

البروتوكول في مجال الشبكات الحاسوبية هو مجموعة من القواعد والقوانين يجري الالتزام بها بهدف إيصال المعلومات إلى الجهة المعنية على نحو صحيح وخالٍ من الأخطاء. ولذا فإن البروتوكول هو بمنزلة اللغة التي تتخاطب بها الأجهزة الحاسوبية المتصلة بالشبكة بهدف تبادل المعلومات ووصولها على النحو الصحيح.

ثمة بروتوكولان مهمان للاتصال بالشبكة، هما:

١ -بروتوكول التحكم في الإرسال (Transmission Control Protocol (TCP : مهمته تفكيك الرزم عند المرسل وتجزئتها، ومن ثم إعادة تجميعها عند المستقبل.

٢ -بروتوكول الإنترنت (Internet Protocol (IP : مهمته إرسال هذه الرزم إلى وجهتها الصحيحة.

يعمل هذان البروتوكولان معاً، ويسمى ذلك بالبروتوكول TCP/IP الذي يوفر خصائص التشبيك networking والتسيير routing ويتيح الاستفادة من موارد الإنترنت.

طُوِّر البروتوكول TCP/IP أساساً في عام ١٩٦٩ من قبل وكالة مشاريع البحوث المتطورة للدفاع الأمريكية، واستُخدم بدايةً لبناء شبكة مشاريع البحوث المتطورة للدفاع الأمريكي، وهي شبكة تربط أربع جامعات أمريكية. ومنذ ذلك الحين أصبح البروتوكول TCP/IP البروتوكول القياسي المستخدم لضمان التوافق بين الأنواع المختلفة من الأجهزة؛ أي هو بروتوكول التحكم في الإرسال وبروتوكول الإنترنت. وحالياً تدعم أغلب الشبكات المحلية والواسعة هذا البروتوكول.

واجهة المقبس عبارة عن شبكات TCP / IP ، أي أنها تحدد مجموعة متنوعة من وظائف أو إجراءات البرامج لتطوير التطبيقات لشبكات TCP / IP. قام مصممو واجهة المقبس في الأصل ببناء واجهتهم في نظام التشغيل UNIX. ومع ذلك ، فإن أنظمة التشغيل والبيئات الأخرى ، مثل Microsoft Windows ، تقوم بتنفيذ واجهة المقبس كمكتبات برامج. ومع ذلك، بغض النظر عن البيئة التي نبرمج فيها، سيبدو رمز البرنامج متماثلاً إلى حد كبير. يمكن إجراء برمجة المقبس بأي لغة، مما يدعم اتصال الشبكة، ولكن يفضل استخدام بايثون لأنه مستقل عن النظام الأساسي، وله آليات استثناء للتعامل القوي مع المشكلات الشائعة التي تحدث أثناء عمليات الإدخال / الإخراج والشبكات ، وتوفر مرافق الترابط الخاصة به طريقة لتنفيذ خوادم قوية بسهولة تتمثل إحدى تطبيقات بايثون الرئيسية القوية كلغة برمجة لتطوير تطبيقات خادم العميل في النطاق الواسع من دعم الشبكة. تتمتع بايثون بهذه الميزة لأنه تم تطويرها مع وضع الإنترنت في الاعتبار. ميزة أخرى لبايثون هي أنها توفر الأمان. والنتيجة هي أن لدينا الكثير من الخيارات فيما يتعلق ببرمجة الشبكة في بايثون. تقوم بايثون بتنفيذ جميع اتصالات الشبكة الخاصة بها من خلال مآخذ التوصيل.

برمجة المقبس هي طريقة لربط عقدتين على شبكة للتواصل مع بعضهما البعض. يستمع أحد المقبس (العقدة) إلى منفذ معين على IP ، بينما يصل المقبس الآخر إلى الآخر لتشكيل اتصال. يشكل الخادم مقبس المستمع بينما يصل العميل إلى الخادم.

## أهمية البحث وأهدافه:

تتبع أهمية البحث من التطور الكبير في عالم التواصل عبر الشبكة. يكون التواصل إما عن طريق النصوص أو الصور أو الفيديو. في الوقت الراهن أصبحت جميع الشركات الكبرى تستخدم التواصل عبر الفيديو من أجل إنشاء الاجتماعات والتواصل مع الزبائن وأيضاً أصبح الإنسان بشكل روتيني في حياته اليومية يفضل التواصل عبر الفيديو مع أقاربه وأصدقائه وبالتالي تصير المسافات بين نقاط مختلفة حول أنحاء العالم. يقدم هذا البحث برنامج محادثة فيديو بين شخصين عبر الشبكة وبالتالي يضيف هذا البحث شرح عن كيفية التعامل مع الفيديو من خلال الشبكة وباستخدام لغة البرمجة بايثون.

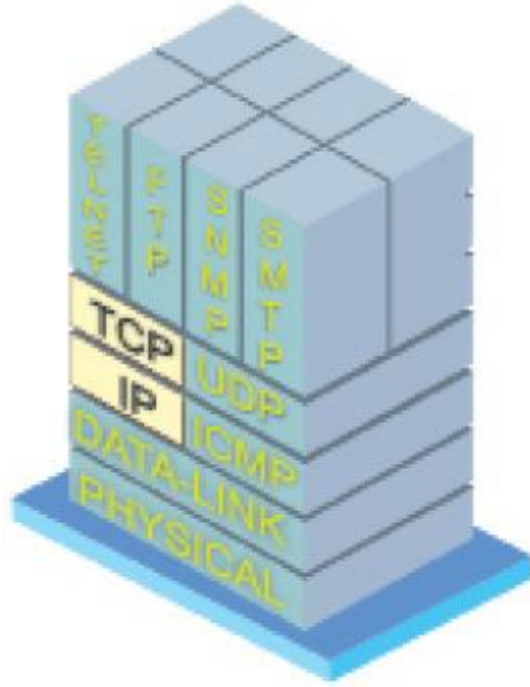
## مواد وطرائق البحث:

- python programming language
- socket library
- open cv library
- numpy library
- threading library
- pycharm

## منهجيات البحث:

### ١ - النموذج TCP/IP:

يُعدّ البروتوكولان TCP و IP من البروتوكولات المحورية في النموذج المرجعي لشبكة الإنترنت كما هو موضح في الشكل (١).



الشكل ١ موقع البروتوكولين في النموذج المرجعي للشبكات

يُخصّص البروتوكول TCP للنقل، ويوفر اتصالاً موجهاً يدعم الاتصال المثنوي الكامل full duplex ويوفّر التحكم في تدفق المعطيات. أما البروتوكول IP فهو بروتوكول شبكة network protocol يوفّر تسليم المعطيات من دون اتصال مسبق؛ إذ تسلك حزم المعطيات مسارات مختلفة بين الحاسوب المرسل والمستقبل في شبكة الإنترنت.

يعتمد البروتوكول TCP/IP على تقنية ابتداء الرزم packet switching التي تدل على تقسيم المعطيات إلى رزم منفصلة وحيدة الهوية. يُعدّ البروتوكول TCP/IP في الجهاز المرسل مسؤولاً عن المهمات التالية:

١ -تقسيم المعطيات إلى رزم.

٢ -إضافة معلومات العنونة إلى الرزم.

٣ -تحضير المعطيات للإرسال.

في حين يقوم البروتوكول TCP/IP في الجهاز المستقبل بالمهمات التالية:

- ١ -التقاط رزم المعطيات من وسط الاتصال.
- ٢ -إدخال رزم المعطيات إلى الحاسوب عبر بطاقة الشبكة.
- ٣ -تجميع كل رزم المعطيات المرسله وقراءة معلومات التحكم المضافة إليها.
- ٤ -نسخ المعطيات من الرزم إلى ذاكرة مؤقتة لإعادة تجميعها.
- ٥- تمرير المعطيات المعاد تجميعها إلى البرامج في صورة مفهومة قابلة للاستخدام.

ولعل من أسباب انتشار البروتوكول TCP/IP هو سماحه للشبكات المستقلة الاتصال بالإنترنت، أو الاتصال ببعضها لتشكيل شبكات إنترانت intranet خاصة. ويوفّر البروتوكول TCP/IP الاتصال بين أي حاسوبين متصلين إلى شبكة تعمل وفقاً لهذا البروتوكول، ويخفي البروتوكول TCP/IP عملياً المسيرّات routers وبنية الشبكة الداخلية ويجعلها تبدو شبكةً كبيرة واحدة.

يُعرف الوصل بالإنترنت بواسطة العنوان IP بطول ٣٢ بتاً، ويُعبّر عن هذا العنوان بواسطة أعداد عشرية منقّطة مثل ١٢٧,٢,٣,٤٤. وعند إسناد عنوان IP إلى حاسوب بعيد، يمكن لحاسوب مرتبط إلى الإنترنت أو الإنترنت إرسال المعطيات إلى ذلك الحاسوب تماماً كما لو أن الحاسوبين هما جزء من الشبكة الفيزيائية نفسها. يقدّم إذاً البروتوكول TCP/IP حلاً لتبادل المعطيات بين حاسوبين متصلين بالإنترنت أو بشبكات فيزيائية مختلفة.

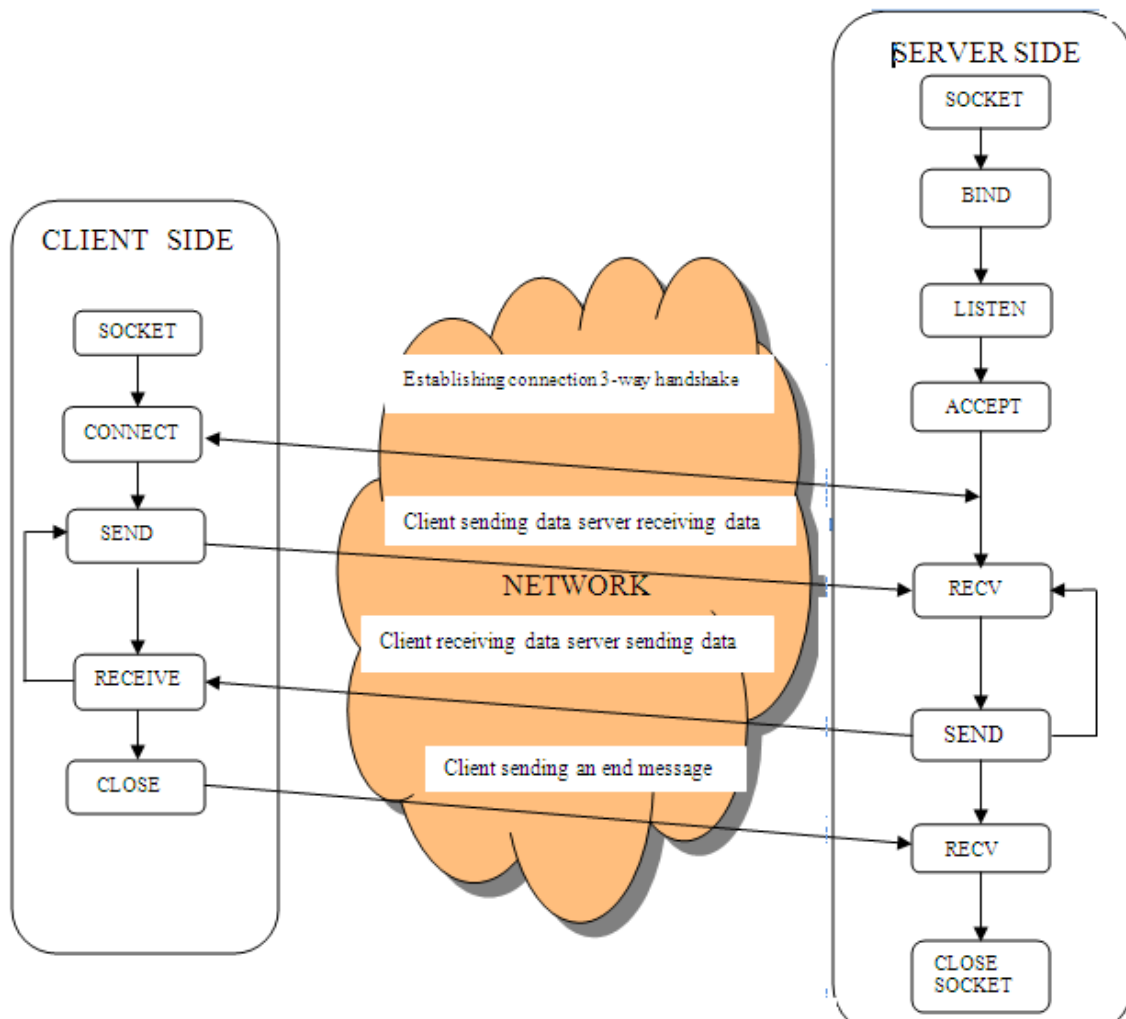
يرسل البروتوكول IP - وهو البروتوكول الأهم في مجموعة بروتوكولات TCP/IP - رزم معطيات IP عبر الشبكة، ويتطلب ذلك القيام بالتسيير لاختيار الممرّ الذي تسلكه رزم المعطيات عبر الشبكة للوصول من جهة إلى أخرى. ويمثل البروتوكول TCP بروتوكول المستوى الأعلى الذي يسمح لتطبيقات منقّدة على حواسيب مضيقة مختلفة بتبادل تدفقات معطيات streams، ويقسم البروتوكول TCP هذه التدفقات إلى كتل تسمى مقاطع TCP segments ويقوم بنقلها مستخدماً البروتوكول IP.

## ٢- مكتبة socket

برمجة المقبس هي طريقة لربط عقدتين على شبكة للتواصل مع بعضهما البعض. يستمع أحد المقبس (العقدة) إلى منفذ معين على IP ، بينما يصل المقبس الآخر إلى الآخر لتشكيل اتصال. يشكل الخادم مقبس المستمع بينما يصل العميل إلى الخادم. هم العمود الفقري الحقيقي وراء تصفح الويب. بعبارة أبسط ، هناك خادم وعميل. تبدأ برمجة Socket عن طريق استيراد مكتبة المقابس وعمل مقبس بسيط.

المأخذ هي نقاط النهاية لقناة اتصالات ثنائية الاتجاه. قد تتواصل المقابس ضمن عملية أو بين عمليات على نفس الجهاز أو بين عمليات في قارات مختلفة. يمكن تنفيذ المآخذ عبر عدد من أنواع القنوات المختلفة: مأخذ مجال Unix و TCP و UDP

وما إلى ذلك. توفر مكتبة المقبس فئات محددة للتعامل مع وسائل النقل الشائعة بالإضافة إلى واجهة عامة للتعامل مع الباقي. المقابس لها مفرداتها الخاصة.



الشكل ٢ آلية عمل وتوابع سوكيت tcp

### ٣- مكتبة open cv

مكتبة OpenCV من أشهر المكتبات المختصة والقوية في مجال معالجة الصور والفيديو، وتتنوع المجالات التي يمكن ان نستفيد فيها من مكتبة OpenCV وخاصة في المجالات الطبية، والصناعية والذكاء الاصطناعي وتعليم الآلة. تتكون OpenCV من عدة خوارزميات وشيفرات برمجية تم جمعها في إطار واحد. الهدف الرئيسي لهذا الاطار هو تقديم حلول في الرؤية الحاسوبية Computer Vision.

يجب في البداية تحميل مكتبة open cv إلى جهاز الحاسوب ويتم ذلك من خلال موجه الأوامر بكتابة السطر التالي:

```
pip install opencv-python
```

### ٤- مكتبة numpy:

مكتبة Numpy من المكتبات التي لا تستطيع تجاهلها عند تعلمك علم البيانات وذلك لسببين رئيسين، الأول هو أن العديد من مكتبات علم البيانات وتعلم الآلة تعتمد اعتمادا قويا عليها، والسبب الثاني هو أن هذه المكتبة تتيح لك قدرة التعامل مع المصفوفات بطريقة أفضل من Lists الموجودة تلقائيًا كشكل من أشكال تراكيب البيانات في البايثون. وكون الصور عبارة عن مصفوفات، فإن هذه المكتبة تقدم فوائد كبيرة في التعامل مع هذه المصفوفات. يجب في البداية تحميل مكتبة numpy إلى جهاز الحاسوب ويتم ذلك من خلال موجه الأوامر بكتابة السطر التالي:

```
pip install numpy
```



## القسم العملي:

أثناء تنفيذ البرنامج يتم تنفيذ جملة واحدة في كل مرة. ولكن لإنشاء تطبيق دردشة فيديو ثنائي الاتجاه ، فإننا نستقبل الصور ونرسلها في نفس الوقت. يتم حل هذه المشكلة من خلال مفهوم الخيوط. هنا يمكننا إرسال أجزاء مختلفة من الكود ليتم تنفيذها في أجزاء مختلفة من وحدة المعالجة المركزية ، من خلال وضعها في سلاسل مختلفة. وبالتالي فإن تنفيذها بشكل متوازي يوفر الوقت ويحسن أداء البرنامج.

أولاً قمنا باستيراد جميع المكتبات المطلوبة.

```
import socket
import cv2, numpy, threading
```

في السطرين التاليين يتم إنشاء مآخذ التوصيل للإرسال والاستلام على التوالي. حيث يمكن استخدام مقيس واحد لغرض واحد فقط.

```
ss = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
cs = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

تُستخدم الوظيفة bind لربط المنفذ المحدد بـ ip ، فهي تأخذ مجموعة من (port n ، ip) كعنوان. بعد ذلك ، التابع Accept يساعد على إنشاء اتصالات من مستخدم آخر. يستمر في الانتظار حتى يحصل على اتصال.

```
ss.bind(("127.0.0.1", 2022))
ss.listen(5)

c_s, addr = ss.accept()
print("Connected to - ", addr)
```

يرسل تابع connect طلباً إلى عنوان IP والمنفذ المعينين للاتصال كعميل.

```
cs.connect(("127.0.0.1", 3033))
cs.settimeout(1)
```

في ملف "app2.py" يتم وضع تابع connect فوق تابع Accept. هذا لأن تابع Accept يوقف تنفيذ البرنامج حتى يحصل على اتصال واحد. لذلك سيتم إنشاء شبكة الاتصال في التسلسل التالي:

- سيرسل البرنامج ٢ طلب الاتصال إلى البرنامج ١.
  - سيقبل البرنامج ١ الطلب.
  - بعد تنفيذ تابع Accept للبرنامج ١ ، ستقوم التابع connect بالبرنامج ١ بتقديم طلب إلى البرنامج ٢.
  - في النهاية التابع Accept من البرنامج ٢ سيقبل طلب البرنامج ١.
- يتم استخدام التابع Settimeout لتعيين وقت تنفيذ للتابع recv ، بحيث لا يتم قطع البرنامج أثناء انتظار recv للبيانات من برنامج آخر.

```
flag = -1
cap = cv2.VideoCapture(0)
```

flag متغير مما يساعد على إغلاق أحد التوابع لأنها تعمل في ثريدات مختلفة. يساعد التابع VideoCapture لإنشاء اتصال من كاميرا الجهاز. هنا البرامتر ٠ من أجل الكاميرا الداخلية و ١ من أجل الكاميرا الخارجية.

```
def send():
    global flag, cap
    while True:
        if flag == 0:
            break
        try:
            ret, photo = cap.read()
            b_img = cv2.imencode(".jpg", photo)[1].tobytes()
            c_s.sendall(b_img)
        except:
            continue
    cap.release()
    ss.close()
```

الآن لدينا التابع send. يستخدم هذا التابع لإرسال الصور إلى خادم آخر باستخدام ss (مقيس الإرسال). وضعنا الكود في حلقة لانتهائية بحيث يستمر التابع في إرسال الصور باستمرار. التابع read يستخدم لالتقاط الصور.

اضطررنا إلى استخدام الكلمة المحجوزة global لأن تابع send سيعمل في ثريد مختلفة. لإرسال أي بيانات عبر الشبكة، يتعين علينا تحويلها إلى تنسيق بايت. في السطر التالي ، يستخدم تابع imencode لتحويل تنسيق الصورة إلى بيانات متدفقة وتعيينها إلى ذاكرة التخزين المؤقت للذاكرة ، ثم يقوم التابع tobytes بتحويل البيانات إلى تنسيق بايت. سيرسل التابع البيانات إلى المقبس المتصل. عندما يكون التابع على وشك الانتهاء، فإن cap.release ستفصل الكاميرا عن البرنامج وسيؤدي ss.close إلى إغلاق المقبس.

```
def receive():
    global flag, cap
    while True:
        try:
            mess = cs.recv(100000)
            if mess:
                nparr = numpy.frombuffer(mess, numpy.uint8)
                img1 = cv2.imdecode(nparr, cv2.IMREAD_COLOR)
```

في التابع receive ، قمنا أيضاً بوضع كل الكود في حلقة لانتهائية. يستخدم التابع cs.recv لتلقي البيانات من cs (مقبس الاستلام).

يعمل التابع frombuffer على تحويل بيانات البايت إلى مصفوفة عددية. ثم يقوم التابع imdecode باستخراج الصورة الأصلية من المصفوفة. أثناء نقل البيانات عبر الشبكة ، قد يتلف جزء ما لهذا السبب وضعنا معظم الكود في try-except حتى لا يتأثر البرنامج بأكمله بصور تالفة واحدة.

```
ret, photo1 = cap.read()
resized = cv2.resize(photo1, (150, 150),
interpolation=cv2.INTER_AREA)
img1[10:160, 10:160] = resized
img1 = cv2.rectangle(img1, (0, 0), (170, 170), (0, 0, 250), 3)
```

في المقطع أعلاه ، يتم التقاط الصور التي يتم إرسالها إلى برنامج آخر وإضافتها إلى الزاوية اليسرى العليا للصور المستلمة. تستخدم طريقة resize () لتغيير شكل وحجم الصورة.

```
cv2.imshow("From server2", img1)
if cv2.waitKey(10) == 13:
    break
except:
    continue
```

```
flag = 0
cv2.destroyAllWindows() # destroy the image window
cs.close()
```

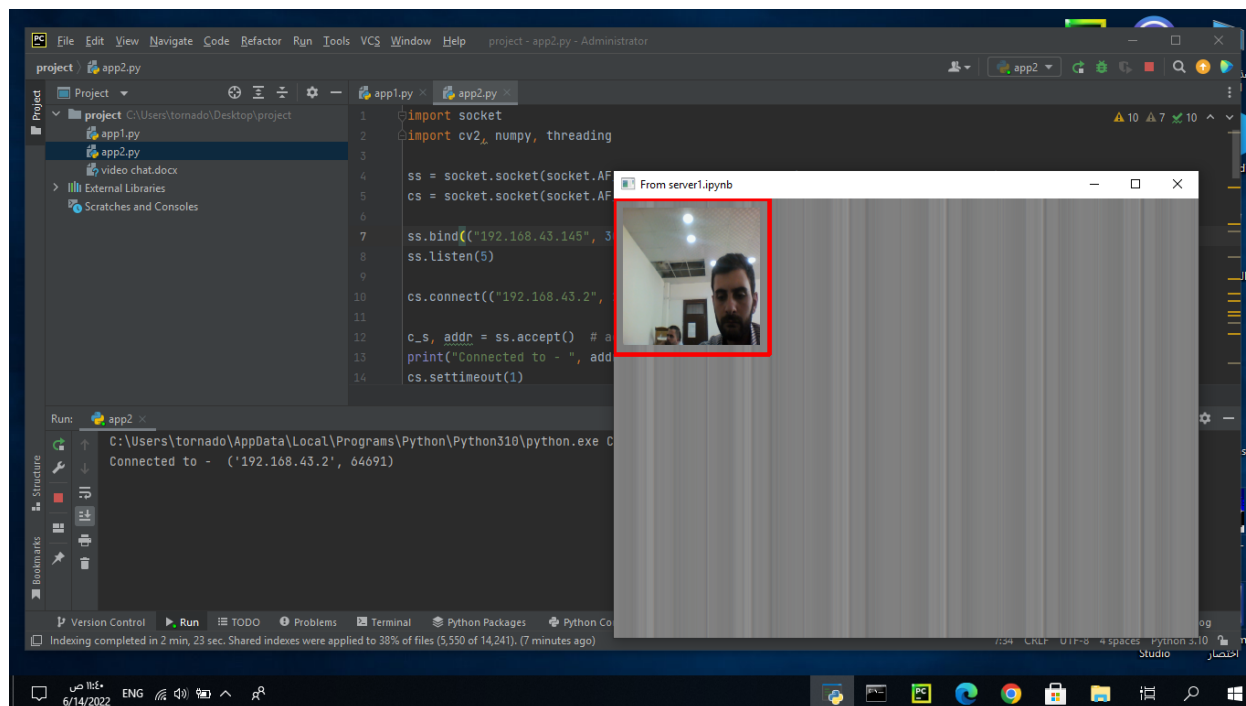
في نهاية تابع الاستقبال ، يستخدم التابع imshow لعرض الصورة النهائية. يلتقط التابع waitKey مفتاح استجابة المستخدم من لوحة المفاتيح، إن وجدت. وهنا سوف يخرج من الحلقة إذا ضغط المستخدم على مفتاح الإدخال enter. في نهاية Receive سيتم تعيين المتغير flag على القيمة ٠ ، وهذا سينهي التابع send إذا كان لا يزال قيد التشغيل. cv2.destroyAllWindows سوف تغلق النافذة التي تم إنشاؤها لعرض الصور.

```
t1 = threading.Thread(target=receive)
t2 = threading.Thread(target=send)
t1.start()
t2.start()
```

في نهاية الكود ، قمنا بوضع توابع الإرسال والاستقبال في ثريدات t1 و t2 . ثم تشغيلهم باستخدام التابع Start. الآن ستستمر الوظيفتان في العمل دون مقاطعة بعضهما البعض.

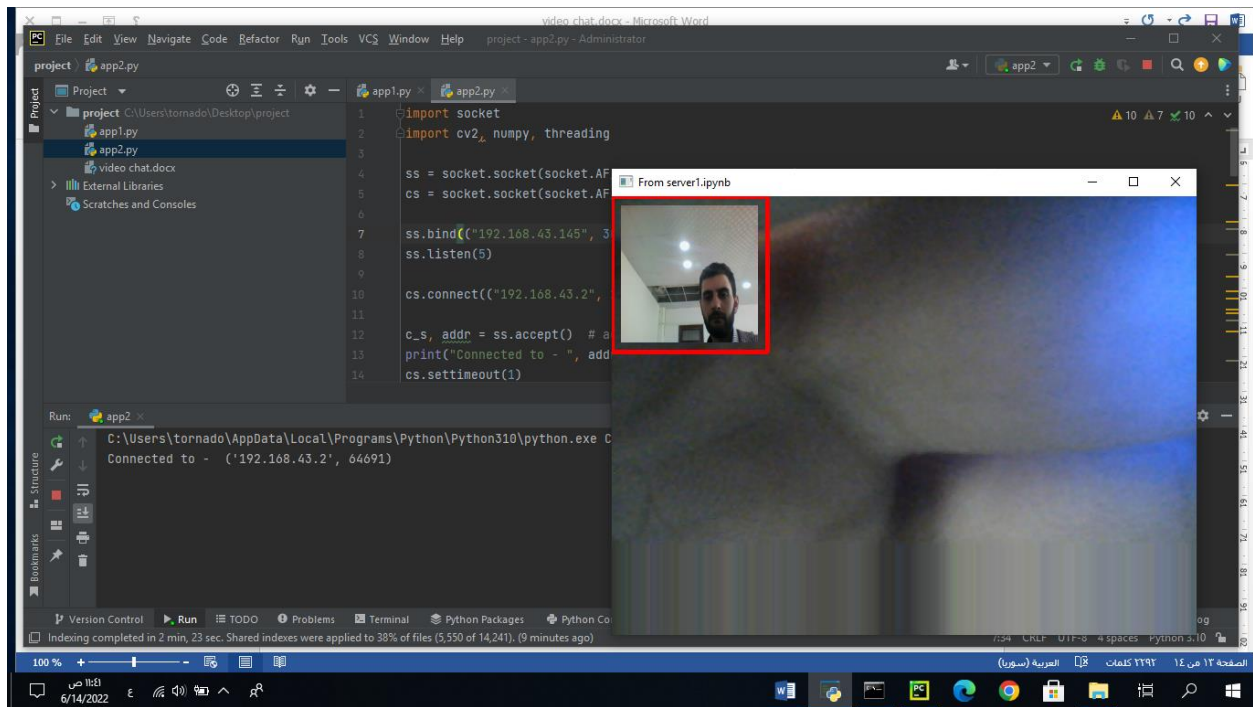
## المناقشة والنتائج:

نقوم باستخراج عناوين ال ip لكل جهاز محمول ونضبط بارمترات التابعين bind و connect وبعدها نقوم بتشغيل البرنامج:



الشكل ١ واجهة البرنامج مع وجود تقطيع بالصورة

نلاحظ في الشكل ١ وجود تقطيع بالصورة وذلك يعود إلى تقنية الترميز المستخدمة في الصورة وضعف الاتصال عبر ال wifi .



الشكل ٢ واجهة البرنامج مع عرض الصورة بشكل صحيح

## الاستنتاجات:

تصف هذه المقالة التفاصيل حول المقابس والمنافذ وبرمجة المقبس عبر TCP . تستخدم برمجة الشبكة المقبس للاتصال بين العمليات المضيفة حيث تعمل المقابس كنقطة نهاية للاتصال بين العمليات. هنا يمكن أيضاً تسمية المقابس على أنها مقبس الشبكة أو مقبس الإنترنت لأن الاتصال بين أجهزة الكمبيوتر يعتمد على بروتوكول الإنترنت. لذا فإن برمجة الشبكة هي أيضاً برمجة Socket Programming. تصف المقالة أيضاً برمجة المقبس في بايثون عبر بروتوكول TCP. ورأينا كيف يتم إرسال الفيديو في الزمن الحقيقي بين جهازين كومبيوتر وذلك بالاستفادة من مفهوم الثريد.

## المراجع:

- [1] Kalita, L. (2014). Socket programming. *International Journal of Computer Science and Information Technologies*, 5(3), 4802-4807.
- [2] Howse, J. (2013). *OpenCV computer vision with python*. Birmingham: Packt Publishing.
- [3] <https://medium.com/geekculture/creating-video-chat-app-using-python-9da0a9c386ba>
- [4] Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1, p. 85). USA: Trelgol Publishing.