# PSET 2

These are some of the libraries/modules you will require for this homework.

```
%load_ext autoreload
%autoreload 2
%matplotlib inline
import os
import sys
!pip install scikit-image
!pip install trulens_eval==0.23.0 chromadb==0.4.18 openai==1.3.7


import numpy as np
import scipy
from PIL import Image
import skimage

from skimage import data
from skimage.transform import warp, AffineTransform
import matplotlib.pyplot as plt
import copy
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
Requirement already satisfied: scikit-image in
/opt/homebrew/lib/python3.11/site-packages (0.22.0)
Requirement already satisfied: numpy>=1.22 in
/opt/homebrew/lib/python3.11/site-packages (from scikit-image)
(1.26.1)
Requirement already satisfied: scipy>=1.8 in
/opt/homebrew/lib/python3.11/site-packages (from scikit-image)
(1.11.3)
Requirement already satisfied: networkx>=2.8 in
/opt/homebrew/lib/python3.11/site-packages (from scikit-image) (3.2.1)
Requirement already satisfied: pillow>=9.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from scikit-image)
(10.1.0)
Requirement already satisfied: imageio>=2.27 in
/opt/homebrew/lib/python3.11/site-packages (from scikit-image)
(2.34.0)
Requirement already satisfied: tifffile>=2022.8.12 in
/opt/homebrew/lib/python3.11/site-packages (from scikit-image)
(2024.2.12)
Requirement already satisfied: packaging>=21 in
/Users/krishpatel/Library/Python/3.11/lib/python/site-packages (from
scikit-image) (23.2)
```

```
Requirement already satisfied: lazy_loader>=0.3 in
/opt/homebrew/lib/python3.11/site-packages (from scikit-image) (0.3)

[notice] A new release of pip is available: 23.3.2 -> 24.0
[notice] To update, run: python3.11 -m pip install --upgrade pip
Requirement already satisfied: trulens_eval==0.23.0 in
/opt/homebrew/lib/python3.11/site-packages (0.23.0)
Requirement already satisfied: chromadb==0.4.18 in
/opt/homebrew/lib/python3.11/site-packages (0.4.18)
Requirement already satisfied: openai==1.3.7 in
/opt/homebrew/lib/python3.11/site-packages (1.3.7)
Requirement already satisfied: numpy>=1.23.5 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(1.26.1)
Requirement already satisfied: frozendict>=2.3.8 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(2.4.0)
Requirement already satisfied: munch>=3.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(4.0.0)
Requirement already satisfied: dill>=0.3.7 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(0.3.8)
Requirement already satisfied: tqdm>=4.66.1 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(4.66.1)
Requirement already satisfied: requests>=2.31.0 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(2.31.0)
Requirement already satisfied: nest-asyncio>=1.5.8 in
/Users/krishpatel/Library/Python/3.11/lib/python/site-packages (from
trulens_eval==0.23.0) (1.5.8)
Requirement already satisfied: typing-extensions>=4.9.0 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(4.9.0)
Requirement already satisfied: python-dotenv>=1.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(1.0.1)
Requirement already satisfied: pydantic<3,>=2 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(2.6.1)
Requirement already satisfied: merkle-json>=1.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(1.0.0)
Requirement already satisfied: langchain>=0.0.354 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(0.1.8)
Requirement already satisfied: langchain-core>=0.1.6 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
```

```
(0.1.24)
Requirement already satisfied: typing-inspect>=0.8.0 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(0.9.0)
Requirement already satisfied: millify>=0.1.1 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(0.1.1)
Requirement already satisfied: humanize>=4.6.0 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(4.9.0)
Requirement already satisfied: streamlit>=1.30.0 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(1.31.1)
Requirement already satisfied: streamlit-aggrid>=0.3.4.post3 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(0.3.4.post3)
Requirement already satisfied: streamlit-extras>=0.2.7 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(0.4.0)
Requirement already satisfied: sqlalchemy>=2.0.19 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(2.0.27)
Requirement already satisfied: alembic>=1.11.2 in
/opt/homebrew/lib/python3.11/site-packages (from trulens_eval==0.23.0)
(1.13.1)
Requirement already satisfied: chroma-hnswlib==0.7.3 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(0.7.3)
Requirement already satisfied: fastapi>=0.95.2 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(0.109.2)
Requirement already satisfied: uvicorn>=0.18.3 in
/opt/homebrew/lib/python3.11/site-packages (from
uvicorn[standard]>=0.18.3->chromadb==0.4.18) (0.27.1)
Requirement already satisfied: posthog>=2.4.0 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(3.4.1)
Requirement already satisfied: pulsar-client>=3.1.0 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(3.4.0)
Requirement already satisfied: onnxruntime>=1.14.1 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(1.17.0)
Requirement already satisfied: opentelemetry-api>=1.2.0 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(1.22.0)
Requirement already satisfied: opentelemetry-exporter-otlp-proto-
grpc>=1.2.0 in /opt/homebrew/lib/python3.11/site-packages (from
chromadb==0.4.18) (1.22.0)
```

```
Requirement already satisfied: opentelemetry-instrumentation-
fastapi>=0.41b0 in /opt/homebrew/lib/python3.11/site-packages (from
chromadb==0.4.18) (0.43b0)
Requirement already satisfied: opentelemetry-sdk>=1.2.0 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(1.22.0)
Requirement already satisfied: tokenizers>=0.13.2 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(0.15.2)
Requirement already satisfied: pypika>=0.48.9 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(0.48.9)
Requirement already satisfied: overrides>=7.3.1 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(7.7.0)
Requirement already satisfied: importlib-resources in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(6.1.1)
Requirement already satisfied: grpcio>=1.58.0 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(1.59.3)
Requirement already satisfied: bcrypt>=4.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(4.1.2)
Requirement already satisfied: typer>=0.9.0 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(0.9.0)
Requirement already satisfied: kubernetes>=28.1.0 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(29.0.0)
Requirement already satisfied: tenacity>=8.2.3 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(8.2.3)
Requirement already satisfied: PyYAML>=6.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(6.0.1)
Requirement already satisfied: mmh3>=4.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from chromadb==0.4.18)
(4.1.0)
Requirement already satisfied: anyio<4,>=3.5.0 in
/opt/homebrew/lib/python3.11/site-packages (from openai==1.3.7)
(3.7.1)
Requirement already satisfied: distro<2,>=1.7.0 in
/opt/homebrew/lib/python3.11/site-packages (from openai==1.3.7)
(1.9.0)
Requirement already satisfied: httpx<1,>=0.23.0 in
/opt/homebrew/lib/python3.11/site-packages (from openai==1.3.7)
(0.26.0)
Requirement already satisfied: sniffio in
```

```
/opt/homebrew/lib/python3.11/site-packages (from openai==1.3.7)
(1.3.0)
Requirement already satisfied: Mako in
/opt/homebrew/lib/python3.11/site-packages (from alembic>=1.11.2-
>trulens_eval==0.23.0) (1.3.2)
Requirement already satisfied: idna>=2.8 in
/opt/homebrew/lib/python3.11/site-packages (from anyio<4,>=3.5.0-
>openai==1.3.7) (3.6)
Requirement already satisfied: starlette<0.37.0,>=0.36.3 in
/opt/homebrew/lib/python3.11/site-packages (from fastapi>=0.95.2-
>chromadb==0.4.18) (0.36.3)
Requirement already satisfied: certifi in
/opt/homebrew/lib/python3.11/site-packages (from httpx<1,>=0.23.0-
>openai==1.3.7) (2023.11.17)
Requirement already satisfied: httpcore==1.* in
/opt/homebrew/lib/python3.11/site-packages (from httpx<1,>=0.23.0-
>openai==1.3.7) (1.0.3)
Requirement already satisfied: h11<0.15,>=0.13 in
/opt/homebrew/lib/python3.11/site-packages (from httpcore==1.*-
>httpx<1,>=0.23.0->openai==1.3.7) (0.14.0)
Requirement already satisfied: six>=1.9.0 in
/opt/homebrew/lib/python3.11/site-packages (from kubernetes>=28.1.0-
>chromadb==0.4.18) (1.16.0)
Requirement already satisfied: python-dateutil>=2.5.3 in
/Users/krishpatel/Library/Python/3.11/lib/python/site-packages (from
kubernetes>=28.1.0->chromadb==0.4.18) (2.8.2)
Requirement already satisfied: google-auth>=1.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from kubernetes>=28.1.0-
>chromadb==0.4.18) (2.24.0)
Requirement already satisfied: websocket-client!=0.40.0,!=0.41.*,!
=0.42.*,>=0.32.0 in /opt/homebrew/lib/python3.11/site-packages (from
kubernetes>=28.1.0->chromadb==0.4.18) (1.7.0)
Requirement already satisfied: requests-oauthlib in
/opt/homebrew/lib/python3.11/site-packages (from kubernetes>=28.1.0-
>chromadb==0.4.18) (1.3.1)
Requirement already satisfied: oauthlib>=3.2.2 in
/opt/homebrew/lib/python3.11/site-packages (from kubernetes>=28.1.0-
>chromadb==0.4.18) (3.2.2)
Requirement already satisfied: urllib3>=1.24.2 in
/opt/homebrew/lib/python3.11/site-packages (from kubernetes>=28.1.0-
>chromadb==0.4.18) (2.1.0)
Requirement already satisfied: aiohttp<4.0.0,>=3.8.3 in
/opt/homebrew/lib/python3.11/site-packages (from langchain>=0.0.354-
>trulens_eval==0.23.0) (3.9.3)
Requirement already satisfied: dataclasses-json<0.7,>=0.5.7 in
/opt/homebrew/lib/python3.11/site-packages (from langchain>=0.0.354-
>trulens_eval==0.23.0) (0.6.4)
Requirement already satisfied: jsonpatch<2.0,>=1.33 in
/opt/homebrew/lib/python3.11/site-packages (from langchain>=0.0.354-
>trulens_eval==0.23.0) (1.33)
Requirement already satisfied: langchain-community<0.1,>=0.0.21 in
/opt/homebrew/lib/python3.11/site-packages (from langchain>=0.0.354-
>trulens_eval==0.23.0) (0.0.21)
Requirement already satisfied: langsmith<0.2.0,>=0.1.0 in
/opt/homebrew/lib/python3.11/site-packages (from langchain>=0.0.354-
>trulens_eval==0.23.0) (0.1.2)
Requirement already satisfied: packaging<24.0,>=23.2 in
/Users/krishpatel/Library/Python/3.11/lib/python/site-packages (from
langchain-core>=0.1.6->trulens_eval==0.23.0) (23.2)
Requirement already satisfied: coloredlogs in
/opt/homebrew/lib/python3.11/site-packages (from onnxruntime>=1.14.1-
>chromadb==0.4.18) (15.0.1)
Requirement already satisfied: flatbuffers in
/opt/homebrew/lib/python3.11/site-packages (from onnxruntime>=1.14.1-
>chromadb==0.4.18) (23.5.26)
Requirement already satisfied: protobuf in
/opt/homebrew/lib/python3.11/site-packages (from onnxruntime>=1.14.1-
>chromadb==0.4.18) (4.23.4)
Requirement already satisfied: sympy in
/opt/homebrew/lib/python3.11/site-packages (from onnxruntime>=1.14.1-
>chromadb==0.4.18) (1.12)
Requirement already satisfied: deprecated>=1.2.6 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
api>=1.2.0->chromadb==0.4.18) (1.2.14)
Requirement already satisfied: importlib-metadata<7.0,>=6.0 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
api>=1.2.0->chromadb==0.4.18) (6.11.0)
Requirement already satisfied: backoff<3.0.0,>=1.10.0 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
exporter-otlp-proto-grpc>=1.2.0->chromadb==0.4.18) (2.2.1)
Requirement already satisfied: googleapis-common-protos~=1.52 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
exporter-otlp-proto-grpc>=1.2.0->chromadb==0.4.18) (1.62.0)
Requirement already satisfied: opentelemetry-exporter-otlp-proto-
common==1.22.0 in /opt/homebrew/lib/python3.11/site-packages (from
opentelemetry-exporter-otlp-proto-grpc>=1.2.0->chromadb==0.4.18)
(1.22.0)
Requirement already satisfied: opentelemetry-proto==1.22.0 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
exporter-otlp-proto-grpc>=1.2.0->chromadb==0.4.18) (1.22.0)
Requirement already satisfied: opentelemetry-instrumentation-
asgi==0.43b0 in /opt/homebrew/lib/python3.11/site-packages (from
opentelemetry-instrumentation-fastapi>=0.41b0->chromadb==0.4.18)
(0.43b0)
Requirement already satisfied: opentelemetry-instrumentation==0.43b0
in /opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
instrumentation-fastapi>=0.41b0->chromadb==0.4.18) (0.43b0)
Requirement already satisfied: opentelemetry-semantic-
```

```
conventions==0.43b0 in /opt/homebrew/lib/python3.11/site-packages
(from opentelemetry-instrumentation-fastapi>=0.41b0->chromadb==0.4.18)
(0.43b0)
Requirement already satisfied: opentelemetry-util-http==0.43b0 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
instrumentation-fastapi>=0.41b0->chromadb==0.4.18) (0.43b0)
Requirement already satisfied: setuptools>=16.0 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
instrumentation==0.43b0->opentelemetry-instrumentation-
fastapi>=0.41b0->chromadb==0.4.18) (68.2.2)
Requirement already satisfied: wrapt<2.0.0,>=1.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
instrumentation==0.43b0->opentelemetry-instrumentation-
fastapi>=0.41b0->chromadb==0.4.18) (1.14.1)
Requirement already satisfied: asgiref~=3.0 in
/opt/homebrew/lib/python3.11/site-packages (from opentelemetry-
instrumentation-asgi==0.43b0->opentelemetry-instrumentation-
fastapi>=0.41b0->chromadb==0.4.18) (3.7.2)
Requirement already satisfied: monotonic>=1.5 in
/opt/homebrew/lib/python3.11/site-packages (from posthog>=2.4.0-
>chromadb==0.4.18) (1.6)
Requirement already satisfied: annotated-types>=0.4.0 in
/opt/homebrew/lib/python3.11/site-packages (from pydantic<3,>=2-
>trulens_eval==0.23.0) (0.6.0)
Requirement already satisfied: pydantic-core==2.16.2 in
/opt/homebrew/lib/python3.11/site-packages (from pydantic<3,>=2-
>trulens_eval==0.23.0) (2.16.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/opt/homebrew/lib/python3.11/site-packages (from requests>=2.31.0-
>trulens_eval==0.23.0) (3.3.2)
Requirement already satisfied: altair<6,>=4.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (5.2.0)
Requirement already satisfied: blinker<2,>=1.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (1.7.0)
Requirement already satisfied: cachetools<6,>=4.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (5.3.2)
Requirement already satisfied: click<9,>=7.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (8.1.7)
Requirement already satisfied: pandas<3,>=1.3.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (2.1.3)
Requirement already satisfied: pillow<11,>=7.1.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (10.1.0)
Requirement already satisfied: pyarrow>=7.0 in
```

```
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (15.0.0)
Requirement already satisfied: rich<14,>=10.14.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (13.7.0)
Requirement already satisfied: toml<2,>=0.10.1 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (0.10.2)
Requirement already satisfied: tzlocal<6,>=1.1 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (5.2)
Requirement already satisfied: validators<1,>=0.2 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (0.22.0)
Requirement already satisfied: gitpython!=3.1.19,<4,>=3.0.7 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (3.1.42)
Requirement already satisfied: pydeck<1,>=0.8.0b4 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit>=1.30.0-
>trulens_eval==0.23.0) (0.8.1b0)
Requirement already satisfied: tornado<7,>=6.0.3 in
/Users/krishpatel/Library/Python/3.11/lib/python/site-packages (from
streamlit>=1.30.0->trulens_eval==0.23.0) (6.3.3)
Requirement already satisfied: python-decouple<4.0,>=3.6 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
aggrid>=0.3.4.post3->trulens_eval==0.23.0) (3.8)
Requirement already satisfied: entrypoints>=0.4 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (0.4)
Requirement already satisfied: htbuilder>=0.6.2 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (0.6.2)
Requirement already satisfied: markdownlit>=0.0.5 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (0.0.7)
Requirement already satisfied: prometheus-client>=0.14.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (0.20.0)
Requirement already satisfied: st-annotated-text>=3.0.0 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (4.0.1)
Requirement already satisfied: streamlit-camera-input-live>=0.2.0
in /opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (0.2.0)
Requirement already satisfied: streamlit-card>=0.0.4 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (1.0.0)
Requirement already satisfied: streamlit-embedcode>=0.1.2 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (0.1.2)
Requirement already satisfied: streamlit-faker>=0.0.2 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (0.0.3)
Requirement already satisfied: streamlit-image-
coordinates<0.2.0,>=0.1.1 in /opt/homebrew/lib/python3.11/site-
packages (from streamlit-extras>=0.2.7->trulens_eval==0.23.0) (0.1.6)
Requirement already satisfied: streamlit-keyup>=0.1.9 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (0.2.3)
Requirement already satisfied: streamlit-toggle-switch>=1.0.2 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (1.0.2)
Requirement already satisfied: streamlit-vertical-slider>=2.5.5 in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (2.5.5)
Requirement already satisfied: huggingface_hub<1.0,>=0.16.4 in
/opt/homebrew/lib/python3.11/site-packages (from tokenizers>=0.13.2-
>chromadb==0.4.18) (0.20.3)
Requirement already satisfied: mypy-extensions>=0.3.0 in
/opt/homebrew/lib/python3.11/site-packages (from typing-
inspect>=0.8.0->trulens_eval==0.23.0) (1.0.0)
Requirement already satisfied: httptools>=0.5.0 in
/opt/homebrew/lib/python3.11/site-packages (from
uvicorn[standard]>=0.18.3->chromadb==0.4.18) (0.6.1)
Requirement already satisfied: uvloop!=0.15.0,!=0.15.1,>=0.14.0 in
/opt/homebrew/lib/python3.11/site-packages (from
uvicorn[standard]>=0.18.3->chromadb==0.4.18) (0.19.0)
Requirement already satisfied: watchfiles>=0.13 in
/opt/homebrew/lib/python3.11/site-packages (from
uvicorn[standard]>=0.18.3->chromadb==0.4.18) (0.21.0)
Requirement already satisfied: websockets>=10.4 in
/opt/homebrew/lib/python3.11/site-packages (from
uvicorn[standard]>=0.18.3->chromadb==0.4.18) (12.0)
Requirement already satisfied: aiosignal>=1.1.2 in
/opt/homebrew/lib/python3.11/site-packages (from
aiohttp<4.0.0,>=3.8.3->langchain>=0.0.354->trulens_eval==0.23.0)
(1.3.1)
Requirement already satisfied: attrs>=17.3.0 in
/opt/homebrew/lib/python3.11/site-packages (from
aiohttp<4.0.0,>=3.8.3->langchain>=0.0.354->trulens_eval==0.23.0)
(23.2.0)
Requirement already satisfied: frozenlist>=1.1.1 in
/opt/homebrew/lib/python3.11/site-packages (from
aiohttp<4.0.0,>=3.8.3->langchain>=0.0.354->trulens_eval==0.23.0)
(1.4.1)
Requirement already satisfied: multidict<7.0,>=4.5 in
/opt/homebrew/lib/python3.11/site-packages (from
aiohttp<4.0.0,>=3.8.3->langchain>=0.0.354->trulens_eval==0.23.0)
```

```
(6.0.5)
Requirement already satisfied: yarl<2.0,>=1.0 in
/opt/homebrew/lib/python3.11/site-packages (from
aiohttp<4.0.0,>=3.8.3->langchain>=0.0.354->trulens_eval==0.23.0)
(1.9.4)
Requirement already satisfied: jinja2 in
/opt/homebrew/lib/python3.11/site-packages (from altair<6,>=4.0-
>streamlit>=1.30.0->trulens_eval==0.23.0) (3.1.3)
Requirement already satisfied: jsonschema>=3.0 in
/opt/homebrew/lib/python3.11/site-packages (from altair<6,>=4.0-
>streamlit>=1.30.0->trulens_eval==0.23.0) (4.21.1)
Requirement already satisfied: toolz in
/opt/homebrew/lib/python3.11/site-packages (from altair<6,>=4.0-
>streamlit>=1.30.0->trulens_eval==0.23.0) (0.12.1)
Requirement already satisfied: marshmallow<4.0.0,>=3.18.0 in
/opt/homebrew/lib/python3.11/site-packages (from dataclasses-
json<0.7,>=0.5.7->langchain>=0.0.354->trulens_eval==0.23.0) (3.20.2)
Requirement already satisfied: gitdb<5,>=4.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from gitpython!
=3.1.19,<4,>=3.0.7->streamlit>=1.30.0->trulens_eval==0.23.0) (4.0.11)
Requirement already satisfied: pyasn1-modules>=0.2.1 in
/opt/homebrew/lib/python3.11/site-packages (from google-auth>=1.0.1-
>kubernetes>=28.1.0->chromadb==0.4.18) (0.3.0)
Requirement already satisfied: rsa<5,>=3.1.4 in
/opt/homebrew/lib/python3.11/site-packages (from google-auth>=1.0.1-
>kubernetes>=28.1.0->chromadb==0.4.18) (4.9)
Requirement already satisfied: more-itertools in
/opt/homebrew/lib/python3.11/site-packages (from htbuilder>=0.6.2-
>streamlit-extras>=0.2.7->trulens_eval==0.23.0) (10.2.0)
Requirement already satisfied: filelock in
/opt/homebrew/lib/python3.11/site-packages (from
huggingface_hub<1.0,>=0.16.4->tokenizers>=0.13.2->chromadb==0.4.18)
(3.13.1)
Requirement already satisfied: fsspec>=2023.5.0 in
/opt/homebrew/lib/python3.11/site-packages (from
huggingface_hub<1.0,>=0.16.4->tokenizers>=0.13.2->chromadb==0.4.18)
(2024.2.0)
Requirement already satisfied: zipp>=0.5 in
/opt/homebrew/lib/python3.11/site-packages (from importlib-
metadata<7.0,>=6.0->opentelemetry-api>=1.2.0->chromadb==0.4.18)
(3.17.0)
Requirement already satisfied: jsonpointer>=1.9 in
/opt/homebrew/lib/python3.11/site-packages (from jsonpatch<2.0,>=1.33-
>langchain>=0.0.354->trulens_eval==0.23.0) (2.4)
Requirement already satisfied: markdown in
/opt/homebrew/lib/python3.11/site-packages (from markdownlit>=0.0.5-
>streamlit-extras>=0.2.7->trulens_eval==0.23.0) (3.5.1)
Requirement already satisfied: lxml in
/opt/homebrew/lib/python3.11/site-packages (from markdownlit>=0.0.5-
>streamlit-extras>=0.2.7->trulens_eval==0.23.0) (5.1.0)
Requirement already satisfied: favicon in
/opt/homebrew/lib/python3.11/site-packages (from markdownlit>=0.0.5-
>streamlit-extras>=0.2.7->trulens_eval==0.23.0) (0.7.0)
Requirement already satisfied: pymdown-extensions in
/opt/homebrew/lib/python3.11/site-packages (from markdownlit>=0.0.5-
>streamlit-extras>=0.2.7->trulens_eval==0.23.0) (10.7)
Requirement already satisfied: pytz>=2020.1 in
/opt/homebrew/lib/python3.11/site-packages (from pandas<3,>=1.3.0-
>streamlit>=1.30.0->trulens_eval==0.23.0) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in
/opt/homebrew/lib/python3.11/site-packages (from pandas<3,>=1.3.0-
>streamlit>=1.30.0->trulens_eval==0.23.0) (2023.3)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/opt/homebrew/lib/python3.11/site-packages (from rich<14,>=10.14.0-
>streamlit>=1.30.0->trulens_eval==0.23.0) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/Users/krishpatel/Library/Python/3.11/lib/python/site-packages (from
rich<14,>=10.14.0->streamlit>=1.30.0->trulens_eval==0.23.0) (2.16.1)
Requirement already satisfied: faker in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
faker>=0.0.2->streamlit-extras>=0.2.7->trulens_eval==0.23.0) (23.2.1)
Requirement already satisfied: matplotlib in
/opt/homebrew/lib/python3.11/site-packages (from streamlit-
faker>=0.0.2->streamlit-extras>=0.2.7->trulens_eval==0.23.0) (3.8.0)
Requirement already satisfied: humanfriendly>=9.1 in
/opt/homebrew/lib/python3.11/site-packages (from coloredlogs-
>onnxruntime>=1.14.1->chromadb==0.4.18) (10.0)
Requirement already satisfied: MarkupSafe>=0.9.2 in
/opt/homebrew/lib/python3.11/site-packages (from Mako-
>alembic>=1.11.2->trulens_eval==0.23.0) (2.1.3)
Requirement already satisfied: mpmath>=0.19 in
/opt/homebrew/lib/python3.11/site-packages (from sympy-
>onnxruntime>=1.14.1->chromadb==0.4.18) (1.3.0)
Requirement already satisfied: smmap<6,>=3.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from gitdb<5,>=4.0.1-
>gitpython!=3.1.19,<4,>=3.0.7->streamlit>=1.30.0-
>trulens_eval==0.23.0) (5.0.1)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/opt/homebrew/lib/python3.11/site-packages (from jsonschema>=3.0-
>altair<6,>=4.0->streamlit>=1.30.0->trulens_eval==0.23.0) (2023.12.1)
Requirement already satisfied: referencing>=0.28.4 in
/opt/homebrew/lib/python3.11/site-packages (from jsonschema>=3.0-
>altair<6,>=4.0->streamlit>=1.30.0->trulens_eval==0.23.0) (0.32.1)
Requirement already satisfied: rpds-py>=0.7.1 in
/opt/homebrew/lib/python3.11/site-packages (from jsonschema>=3.0-
>altair<6,>=4.0->streamlit>=1.30.0->trulens_eval==0.23.0) (0.17.1)
Requirement already satisfied: mdurl~=0.1 in
/opt/homebrew/lib/python3.11/site-packages (from markdown-it-
py>=2.2.0->rich<14,>=10.14.0->streamlit>=1.30.0->trulens_eval==0.23.0)
```

```
(0.1.2)
Requirement already satisfied: pyasn1<0.6.0,>=0.4.6 in
/opt/homebrew/lib/python3.11/site-packages (from pyasn1-
modules>=0.2.1->google-auth>=1.0.1->kubernetes>=28.1.0-
>chromadb==0.4.18) (0.5.1)
Requirement already satisfied: beautifulsoup4>=4.7.0 in
/opt/homebrew/lib/python3.11/site-packages (from favicon-
>markdownlit>=0.0.5->streamlit-extras>=0.2.7->trulens_eval==0.23.0)
(4.12.2)
Requirement already satisfied: contourpy>=1.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from matplotlib-
>streamlit-faker>=0.0.2->streamlit-extras>=0.2.7-
>trulens_eval==0.23.0) (1.1.1)
Requirement already satisfied: cycler>=0.10 in
/opt/homebrew/lib/python3.11/site-packages (from matplotlib-
>streamlit-faker>=0.0.2->streamlit-extras>=0.2.7-
>trulens_eval==0.23.0) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in
/opt/homebrew/lib/python3.11/site-packages (from matplotlib-
>streamlit-faker>=0.0.2->streamlit-extras>=0.2.7-
>trulens_eval==0.23.0) (4.43.1)
Requirement already satisfied: kiwisolver>=1.0.1 in
/opt/homebrew/lib/python3.11/site-packages (from matplotlib-
>streamlit-faker>=0.0.2->streamlit-extras>=0.2.7-
>trulens_eval==0.23.0) (1.4.5)
Requirement already satisfied: pyparsing>=2.3.1 in
/opt/homebrew/lib/python3.11/site-packages (from matplotlib-
>streamlit-faker>=0.0.2->streamlit-extras>=0.2.7-
>trulens_eval==0.23.0) (3.1.1)
Requirement already satisfied: soupsieve>1.2 in
/opt/homebrew/lib/python3.11/site-packages (from
beautifulsoup4>=4.7.0->favicon->markdownlit>=0.0.5->streamlit-
extras>=0.2.7->trulens_eval==0.23.0) (2.5)

[notice] A new release of pip is available: 23.3.2 -> 24.0
[notice] To update, run: python3.11 -m pip install --upgrade pip
```

These are some functions which will be useful throught the homework to (1) display a single grayscale image, (2) display multiple images using subplots.

```python
def display_gray(x: np.array, normalized:bool = False):
    plt.figure(figsize=(10,10))
    if not normalized:
        plt.imshow(x,cmap='gray',vmin=0,vmax=1)
    else:
        plt.imshow(x/x.max(),cmap='gray',vmin=0,vmax=1)

def display_axis(ax: plt.axis, x: np.array, title: str,
normalized:bool = False):
    if not normalized:
        ax.imshow(x,cmap='gray',vmin=0,vmax=1)
    else:
        ax.imshow(x/x.max(),cmap='gray',vmin=0,vmax=1)
    ax.set_title(title,size=18)

def display_axis_bw(ax: plt.axis, x: np.array, title: str,
normalized:bool = False, fig = None):
    if not normalized:
        imax = ax.imshow(x,cmap='gray',vmin=0,vmax=1)
    else:
        imax = ax.imshow(x/x.max(),cmap='gray',vmin=0,vmax=1)
    ax.set_title(title,size=18)
    fig.colorbar(imax,ax=ax)

def display_axis_color(ax: plt.axis, x: np.array, title: str,
normalized:bool = False, fig = None):
    if not normalized:
        imax = ax.imshow(x,cmap='coolwarm',vmin=0,vmax=1)
    else:
        imax = ax.imshow(x/x.max(),cmap='coolwarm',vmin=0,vmax=1)
    ax.set_title(title,size=18)
    fig.colorbar(imax,ax=ax)
```

# Question 2

## Blob Detection

In this question, you will be using the Laplacian of Gaussian Filter to perform blob detection. Using the previous parts of the question you should have an analytical expression for the Laplacian of Gaussian. You will be using that result to design a LoG filter. You will then be using that filter to detect blobs of different scales in the image by varying the standard deviation parameter.

```
# Copy paste your conv2D function from the previous homework here.
'''def conv2D(image: np.array, kernel: np.array = None):
# Zero padding
  #rewriting the function to vectorize using numpy
  height, length = kernel.shape
  img_h, img_w = image.shape
  pad_height = height // 2
  pad_width = length // 2
  padded_img = np.pad(image, ((pad_height, pad_height),(pad_width,
pad_width)), mode='constant')
  result = np.zeros((img_h, img_w))'''

#faster implementation
```

```python
def conv2D(image: np.array, kernel: np.array = None):
    kernel = np.flip(kernel, axis=(0, 1))
    # Zero padding
    height, length = kernel.shape
    pad_height = height // 2
    pad_width = length // 2
    padded_img = np.pad(image, ((pad_height, pad_height), (pad_width,
pad_width)), mode='constant')

    img_h, img_w = image.shape
    result = np.zeros((img_h, img_w))

    for x in range(img_h):
        for y in range(img_w):
            region = padded_img[x:x+height, y:y+length]
            result[x, y] = np.sum(region * kernel)
    return result
'''

def conv2D(image: np.array, kernel: np.array = None):
# Zero padding
  height, length = kernel.shape
  img_h, img_w = image.shape
  pad_height = height // 2
  pad_width = length // 2
  padded_img = np.pad(image, ((pad_height, pad_height),(pad_width,
pad_width)), mode='constant')
  result = np.zeros((img_h, img_w))
  for x in range(img_h):
    for y in range(img_w):
      for i in range(-pad_height, pad_height + 1):
        for j in range(-pad_width, pad_width + 1):
          result[x, y] += padded_img[x+pad_height+i, y+pad_width+j] *
kernel[i+pad_height,j+pad_width]
  return result
  '''
```

```
"\n\ndef conv2D(image: np.array, kernel: np.array = None):\n# Zero
padding\n  height, length = kernel.shape\n  img_h, img_w =
image.shape\n  pad_height = height // 2\n  pad_width = length // 2\n
padded_img = np.pad(image, ((pad_height, pad_height),(pad_width,
pad_width)), mode='constant')\n  result = np.zeros((img_h, img_w))\n
for x in range(img_h):\n    for y in range(img_w):\n      for i in
range(-pad_height, pad_height + 1):\n        for j in range(-
pad_width, pad_width + 1):\n          result[x, y] +=
padded_img[x+pad_height+i, y+pad_width+j] *
kernel[i+pad_height,j+pad_width]\n  return result \n  "
```

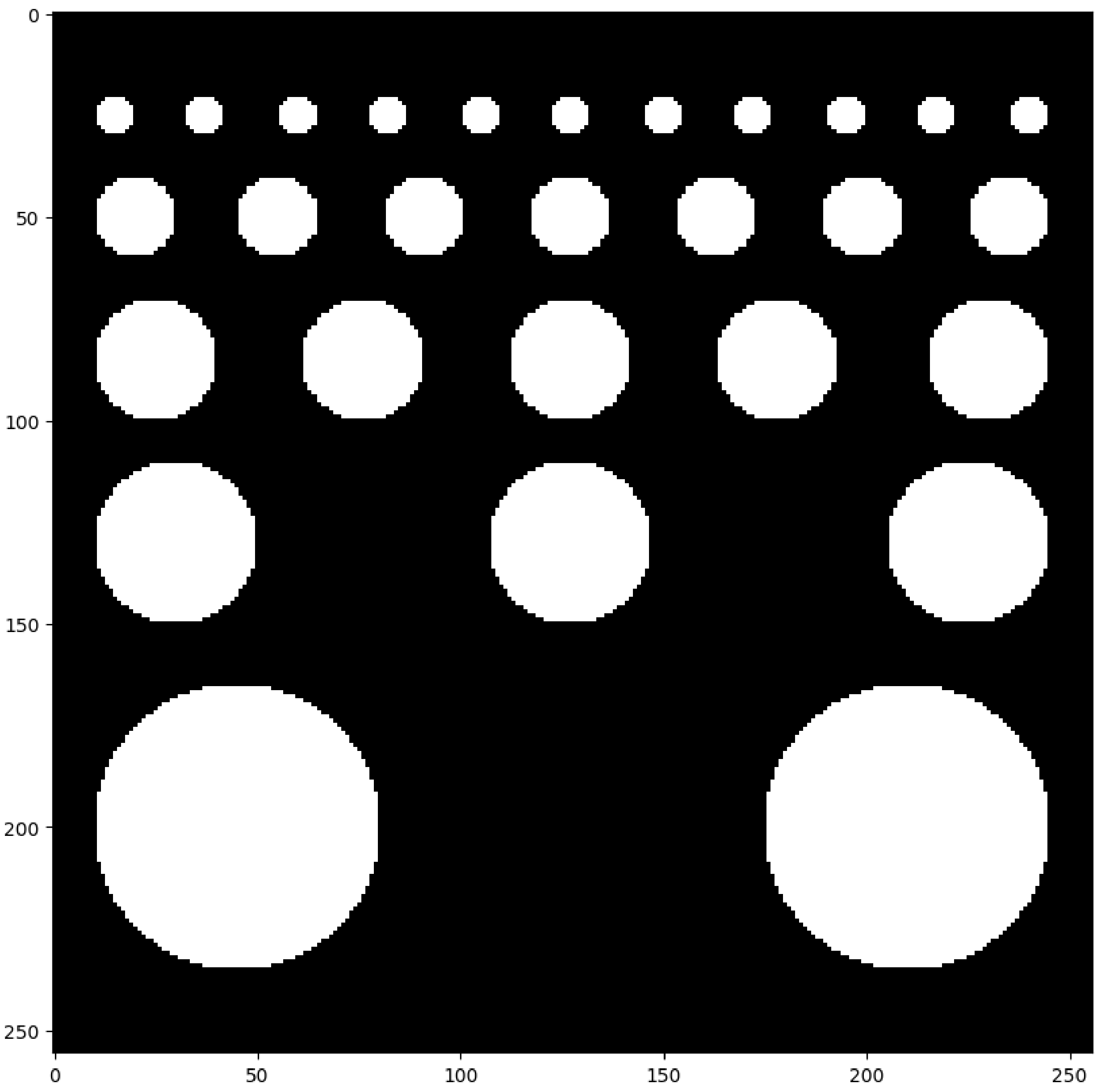
We use the two functions below to create blobs of different sizes. You will have 5 different sizes of blobs in the image below. You will have to tune the standard deviation of your LoG filter such that you get the maximum response.

```python
def make_circle(img: np.array, x: int, y: int, radius: int):
    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            if np.sqrt((x-i)**2 + (y-j)**2) < 1.0*radius:
                img[i,j]=1
    return img

def draw_circle(img, y, radius):
    rad_buffer = radius + 5
    start = rad_buffer+5
    end = 255 - rad_buffer - 5
    centers = np.linspace(start, end, int((end-start)/(2*rad_buffer)))
    for c in centers:
        c = int(c)
        make_circle(img, y, c, radius)
    return img

blob_img = np.zeros((256,256))
blob_img = draw_circle(blob_img, 25, 5)
blob_img = draw_circle(blob_img, 50, 10)
blob_img = draw_circle(blob_img, 85, 15)
blob_img = draw_circle(blob_img, 130, 20)
blob_img = draw_circle(blob_img, 200, 35)

display_gray(blob_img)
```



```
img = copy.deepcopy(blob_img)
```

### Answer 2.5

For this sub-part, you will be writing a function `log_filter(size, sigma)`, which takes as input the size of the LoG filter and the sigma, and returns a scale-normalized LoG filter.

Copy paste your solution in the cell below on Overleaf for Question 2.5.

```
# Write your answer in this cell.
'''def log_filter(size: int, sigma: float):
    x = np.linspace(-size//2, size//2, size)
    y = np.linspace(-size//2, size//2, size)
    x, y = np.meshgrid(x, y)
    kernel = np.exp(-(x**2 + y**2)/(2*sigma**2))
    kernel = kernel / (2 * np.pi * sigma**4) # normalization done here
    return kernel
'''

def log_filter(size: int, sigma: float):
    if size % 2 == 0:
        size += 1
    if size is None:
        size = 6 * sigma
    if size % 2 == 0:
        size += 1
    x, y = np.meshgrid(np.arange(-size//2+1, size//2+1), np.arange(-
size//2+1, size//2+1))
    kernel = -(1/(np.pi * sigma**4)) * (1 - ((x**2 + y**2) / (2 *
sigma**2))) * np.exp(-(x**2 + y**2) / (2 * sigma**2))
    kernel = kernel/np.sum(np.abs(kernel))  # normalization done here
    return kernel
```

As you might have seen above, the image blobs have 5 different scales. Hence you have to find 5 `sigma` values which will give maximum response when the LoG filter is convolved with the image. You may want to use the results from the class lectures to find the values for `sigma`. To visualize maximum response we will be plotting the filtered images using a color map where blue color would correspond to smaller values and red color would correspond to higher values.

### Answer 2.6

Tune the values of the 5 `sigmas`, so that you get the maximum response for each scale. So, `sigma_1` should be such that you get the maximum response for the smallest blobs, `sigma_5` should give you the maximum response for the largest blobs.

Write the values of the 5 `sigmas` you get here on Overleaf.

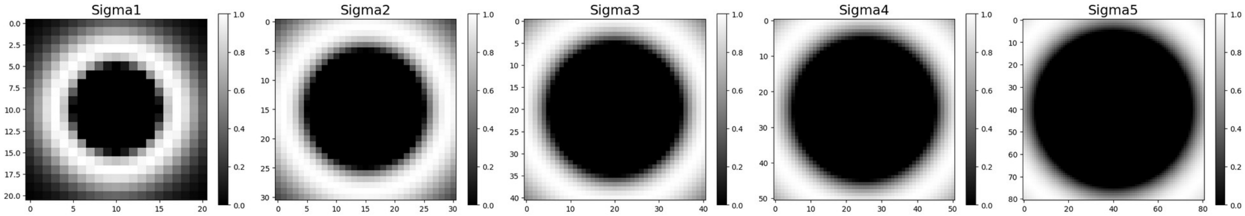
```
sigma_1 = 3.54
log_1 = log_filter(21, sigma_1)
sigma_2 = 7.07
log_2 = log_filter(31, sigma_2)
sigma_3 = 10.61
log_3 = log_filter(41, sigma_3)
sigma_4 = 14.14
log_4 = log_filter(51, sigma_4)
sigma_5 = 24.75
log_5 = log_filter(81, sigma_5)
```

### Answer 2.7

In this sub-part, you will visualize the LoG filters.

Upload the saved image on Overleaf for Question 2.7.

```
fig, ax = plt.subplots(1,5,figsize=(1 + 5*4.5,4))
display_axis_bw(ax[0],log_1,'Sigma1',normalized=True,fig=fig)
display_axis_bw(ax[1],log_2,'Sigma2',normalized=True,fig=fig)
display_axis_bw(ax[2],log_3,'Sigma3',normalized=True,fig=fig)
display_axis_bw(ax[3],log_4,'Sigma4',normalized=True,fig=fig)
display_axis_bw(ax[4],log_5,'Sigma5',normalized=True,fig=fig)
fig.tight_layout()
os.makedirs('Data/Solutions', exist_ok=True)
fig.savefig('Data/Solutions/question_2_7.pdf', format='pdf',
bbox_inches='tight')
```



Convolve the image with the 5 filters. Note that we multiply the filters with -1 so that the maximum response is positive.

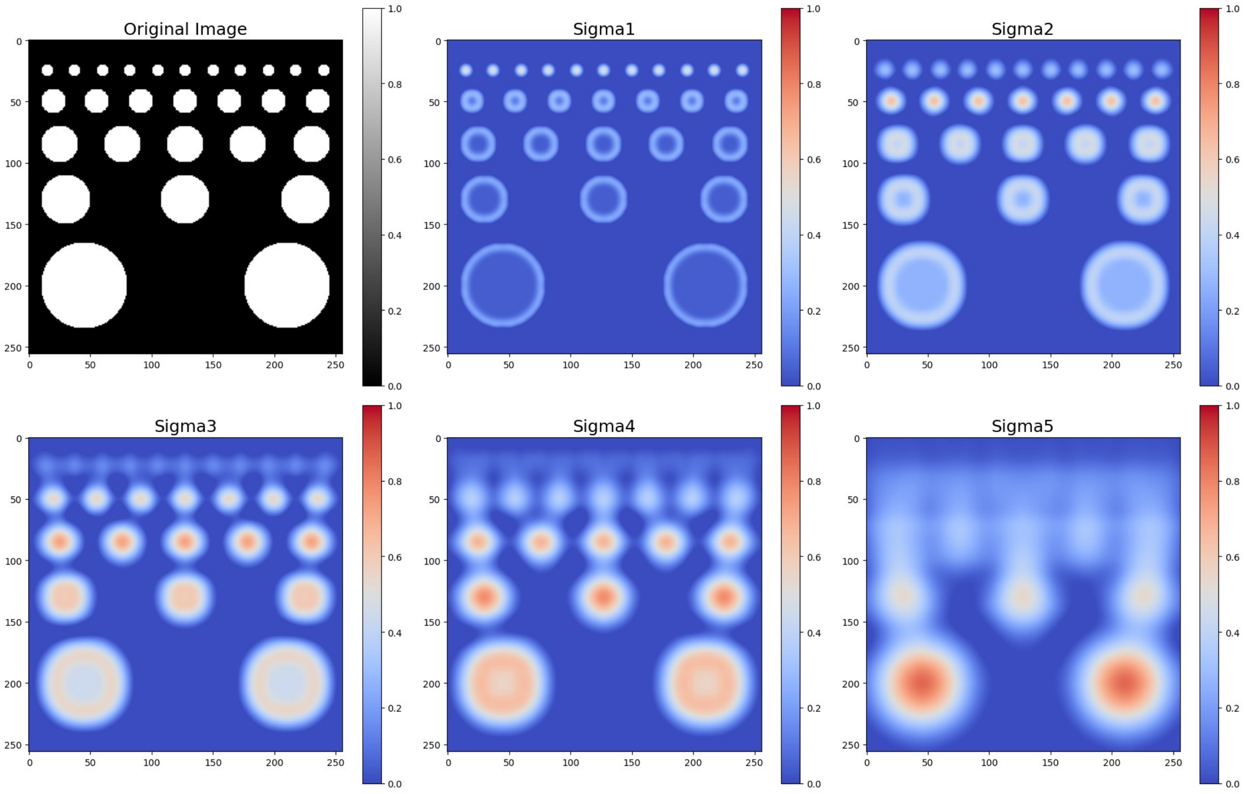
```
log_conv_1 = conv2D(img, -log_1)
log_conv_2 = conv2D(img, -log_2)
log_conv_3 = conv2D(img, -log_3)
log_conv_4 = conv2D(img, -log_4)
log_conv_5 = conv2D(img, -log_5)
```

### Answer 2.8

In this sub-part, you will visualize the blob detection results. We also plot the colorbar with each image. You should use that to tune the values for `sigma`.

Upload the saved image on Overleaf for Question 2.8.

```
fig, ax = plt.subplots(2,3,figsize=(1 + 3*6,2*6))
display_axis_bw(ax[0,0],img,'Original Image',fig=fig)
display_axis_color(ax[0,1],log_conv_1,'Sigma1',fig=fig)
display_axis_color(ax[0,2],log_conv_2,'Sigma2',fig=fig)
display_axis_color(ax[1,0],log_conv_3,'Sigma3',fig=fig)
display_axis_color(ax[1,1],log_conv_4,'Sigma4',fig=fig)
display_axis_color(ax[1,2],log_conv_5,'Sigma5',fig=fig)
fig.tight_layout()
fig.savefig('Data/Solutions/question_2_8.pdf', format='pdf',
bbox_inches='tight')
```



# Question 3

## Corner Detection

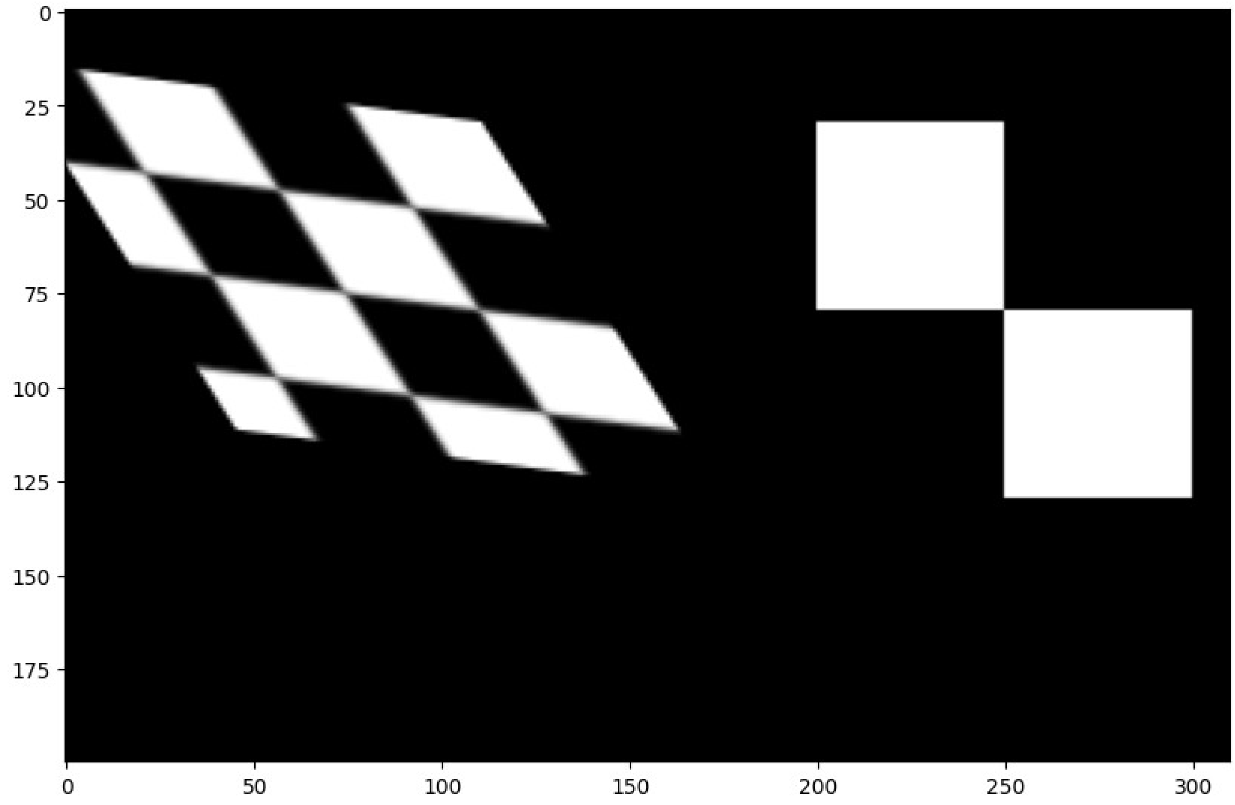
In this question, you will be implementing the Harris corner detector. Corners serve as useful features in images for a variety of reasons: they are salient, well localized, and invariant to a variety of transformations (illumination, rotation, and scale changes). The Harris corner detector also possesses some of these invariances (rotation and intensity shift) but is not invariant to image and intensity scaling. Nevertheless, the Harris corner detector is still a popular method for detecting corners in images.

A grayscale test image (normalized to be in [0, 1]) containing some squares and a warped checkerboard is constructed below.

```
# Sheared checkerboard
tform = AffineTransform(scale=(1.3, 1.1), rotation=1, shear=0.7,
                        translation=(110, 30))
image = warp(data.checkerboard()[:90, :90], tform.inverse,
             output_shape=(200, 310))

# Two squares
image[30:80, 200:250] = 1
image[80:130, 250:300] = 1
```

```
display_gray(image)
```



## Computing Image Gradients

The first step in the Harris corner detector is to compute the image gradients. While there are a variety of different methods to compute gradients, you will use the Sobel filter, which is defined below for the x and y directions.

```
sobel_x = np.array([[1, 0, -1], [2, 0, -2], [1, 0, -1]])
sobel_y = np.array([[1, 2, 1], [0, 0, 0], [-1, -2, -1]])
```

After defining the Sobel filters, you need to apply them to obtain the gradients. Complete a function `compute_image_gradient(image)` that returns the horizontal (along the x direction) and vertical (along the y direction) image gradients using the provided Sobel filters. For this function (and all convolutions that follow), use the `conv2D(image, kernel)` function, which you should copy paste into the cell at the beginning of Question 2.

### Answer 3.1

Copy paste your solution in the cell below on Overleaf for Question 3.1.

```
# Write your code in this cell.
```

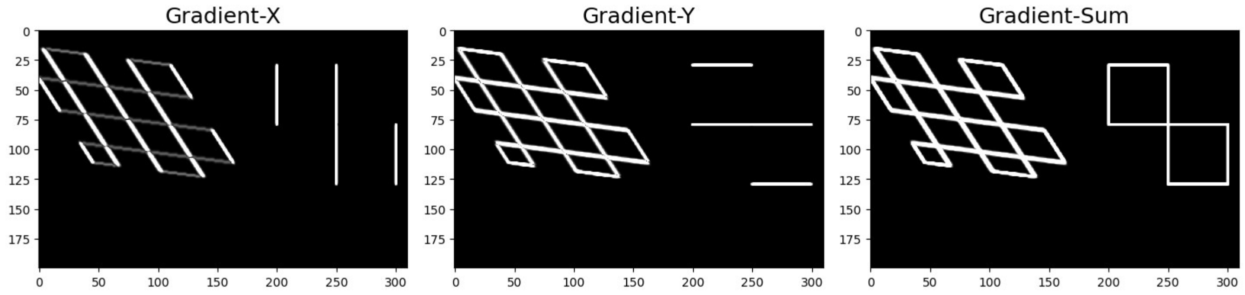
```
def compute_image_gradient(image: np.array):
    sobel_x = np.array([[1, 0, -1], [2, 0, -2], [1, 0, -1]])
    sobel_y = np.array([[1, 2, 1], [0, 0, 0], [-1, -2, -1]])
    grad_x = conv2D(image, sobel_x)
    grad_y = conv2D(image, sobel_y)
    return grad_x, grad_y
```

Display the absolute gradient along the horizontal and vertical directions and their sum. You should observe that the gradient in the horizontal (x-direction) is unable to capture the horizontal edges while the gradient in the vertical (y-direction) is unable to capture the vertical edges. Both gradients capture the diagonal edges, but they may appear dimmer than the horizontal and vertical edges.

### Answer 3.2

Execute the cell below and copy the saved image on Overleaf for Question 3.2.

```
fig, ax = plt.subplots(1,3,figsize=(1 + 3*4.5,4))
img_gradient_x, img_gradient_y = compute_image_gradient(image)
display_axis(ax[0], np.abs(img_gradient_x), 'Gradient-X')
display_axis(ax[1], np.abs(img_gradient_y), 'Gradient-Y')
display_axis(ax[2], np.abs(img_gradient_x) + np.abs(img_gradient_y),
'Gradient-Sum')
fig.tight_layout()
os.makedirs('Data/Solutions', exist_ok=True)
fig.savefig('Data/Solutions/question_3_2.pdf', format='pdf',
bbox_inches='tight')
```



## Computing the Covariance Matrix

After computing the gradients, the Harris corner detector then computes the covariance matrix of the gradients (see lecture 5 slide 26). Complete the function `grad_covariance(image, size)` that computes each pixel's covariance matrix using the `size x size` window centered at the pixel. This function should return three matrices $I_{xx}, I_{xy}, I_{yy}$ containing the top-left, diagonal, and bottom-right terms, respectively, of every pixel's covariance matrix. When computing the covariance, you do not need to subtract the means of the image gradients. The average filter is provided below as a useful function.

```python
# This is the standard box filter which computes the mean of all the pixels inside the filter.
def average_filter(size: int):
    assert size%2 == 1
    return 1.0 * np.ones((size,size))/(size**2)
```

### Answer 3.3

Copy paste your solution in the cell below on Overleaf for Question 3.3.

```python
# Write your code in this cell.

def grad_covariance(image: np.array, size: int):
    avg_filter = average_filter(size)
    grad_x, grad_y = compute_image_gradient(image)
    grad_x_sq = grad_x**2
    grad_y_sq = grad_y**2
    grad_xy = grad_x*grad_y
    grad_x_sq_avg = conv2D(grad_x_sq,avg_filter) # convolve gradient squared with average filter
    grad_y_sq_avg = conv2D(grad_y_sq,avg_filter)
    grad_xy_avg = conv2D(grad_xy,avg_filter)
    return grad_x_sq_avg, grad_y_sq_avg, grad_xy_avg
```

## Harris Response Function

Finally, the Harris corner detector uses the covariance matrix to compute a response function, which is then thresholded to obtain the locations of the corners. Complete the function `harris_response(image, k, size)` which computes the Harris response function (see lecture 5 slide 43, Harris & Stephens (1988)) for an image using a `size x size` window around every pixel. The parameter `k` corresponds to the parameter in the Harris response function.

### Answer 3.4

Copy paste your solution in the cell below on Overleaf for Question 3.4.

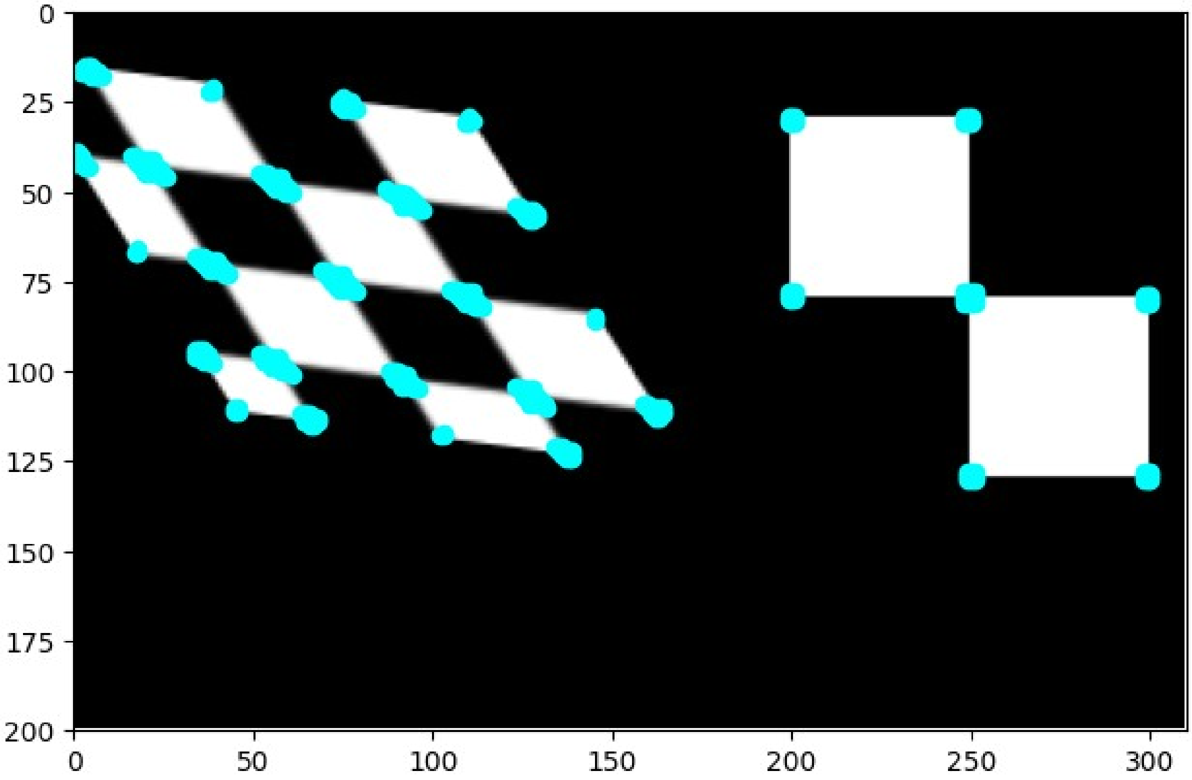
```python
# Write your code in this cell.

def harris_response(image: np.array, k: float, size: int):
    grad_x_sq_avg,grad_y_sq_avg,grad_xy_avg = grad_covariance(image, size)
    det = grad_x_sq_avg * grad_y_sq_avg-grad_xy_avg**2 #determinant
    trace = grad_x_sq_avg+grad_y_sq_avg #trace operator
    return det - k*trace**2
```

### Answer 3.5

Execute the cell below and copy the saved image on Overleaf for Question 3.5. The cell below finds every pixel location where the Harris response function is above a certain threshold and then shows the locations of these pixels (the corner detections) in cyan.

```
coords = np.argwhere(harris_response(image, 0.05, 3) > 0.02)
fig, ax = plt.subplots()
ax.imshow(image, cmap=plt.cm.gray)
ax.plot(coords[:, 1], coords[:, 0], color='cyan', marker='o',
        linestyle='None', markersize=6)
ax.axis((0, 310, 200, 0))
fig.tight_layout()
plt.show()
fig.savefig('Data/Solutions/question_3_5.pdf', format='pdf',
bbox_inches='tight')
```



## Non-Maximum Suppression

You should observe from the above image that all corners are detected. However, some corners are detected multiple times, which is due to simply thresholding the Harris response. To suppress multiple detections, you will implement, in steps, a function `non_max_suppression(harris_response, distance, threshold)` that applies non-maximum suppression to the Harris corner response and returns the remaining corner detections.

Non-maximum suppression works as follows:

1. Threshold the Harris response map to obtain the pixel locations where the response is greater than a certain threshold. These pixel locations form our candidate corner detections.
2. Sort these detections based on maximum response value.
3. Go through the sorted detections in order and for each detection, remove other detections that are within a certain Euclidean distance from the current detection. This step suppresses detections that are not local maxima.

You will implement the three steps of non-maximum suppression in order. First, complete the function `threshold_harris_response(harris_response, threshold)`, which returns the indices of the Harris response map corresponding to values that are greater than some threshold.

### Answer 3.6

Copy paste your solution in the cell below on Overleaf for Question 3.6.

```
# Write your code in this cell.

def threshold_harris_response(harris_response: np.array, threshold:
float):
    return harris_response > threshold
```

Then, complete the function `sort_detections(candidate_detections, harris_response)`, which returns the candidate detections sorted based on maximum Harris response value.

### Answer 3.7

Copy paste your solution in the cell below on Overleaf for Question 3.7.

```
# Write your code in this cell.
def sort_detections(candidate_detections: np.array, harris_response:
np.array):
    sorted_idces = np.argsort(harris_response.flatten())[::-1]
    unravelled_idces = np.unravel_index(sorted_idces,
harris_response.shape)
    sorted_detections = np.column_stack(unravelled_idces)
    index_to_rank = {}
    for rank, idx in enumerate(zip(*sorted_detections)):
        index_to_rank[idx] = rank
    sorted_candidate_detections = sorted(candidate_detections,
key=lambda x: index_to_rank.get(tuple(x), len(candidate_detections)))
    return np.array(sorted_candidate_detections)
```

The final step is to go through the sorted detections and suppress detections that are not local maxima. Complete the function `local_max(sorted_detections, distance)`, which goes through the sorted detections and returns only the detections that are local maxima (using local neighborhoods defined by a Euclidean distance threshold `distance`).

A function that computes Euclidean distance between two points is provided below for convenience.

```
def l2_distance(p1: np.array, p2: np.array):
    return np.linalg.norm(p1 - p2, ord=2)
```

### Answer 3.8

Copy paste your solution in the cell below on Overleaf for Question 3.8.

```
# Write your code in this cell.

def local_max(sorted_detections: np.array, distance: float):
    maxima = []
    for i in range(len(sorted_detections)):
        for j in range(i+1, len(sorted_detections)):
            if
l2_distance(sorted_detections[i],sorted_detections[j])<distance:
                break
        else:
            maxima.append(sorted_detections[i])
    return np.array(maxima)
```

Now, combine the three previously implemented functions to complete the function `non_max_suppression(harris_response, distance, threshold)`, which applies non-maximum suppression to the Harris corner response and returns the remaining corner detections as a NumPy array of (row, col) locations.

### Answer 3.9

Copy paste your solution in the cell below on Overleaf for Question 3.9.

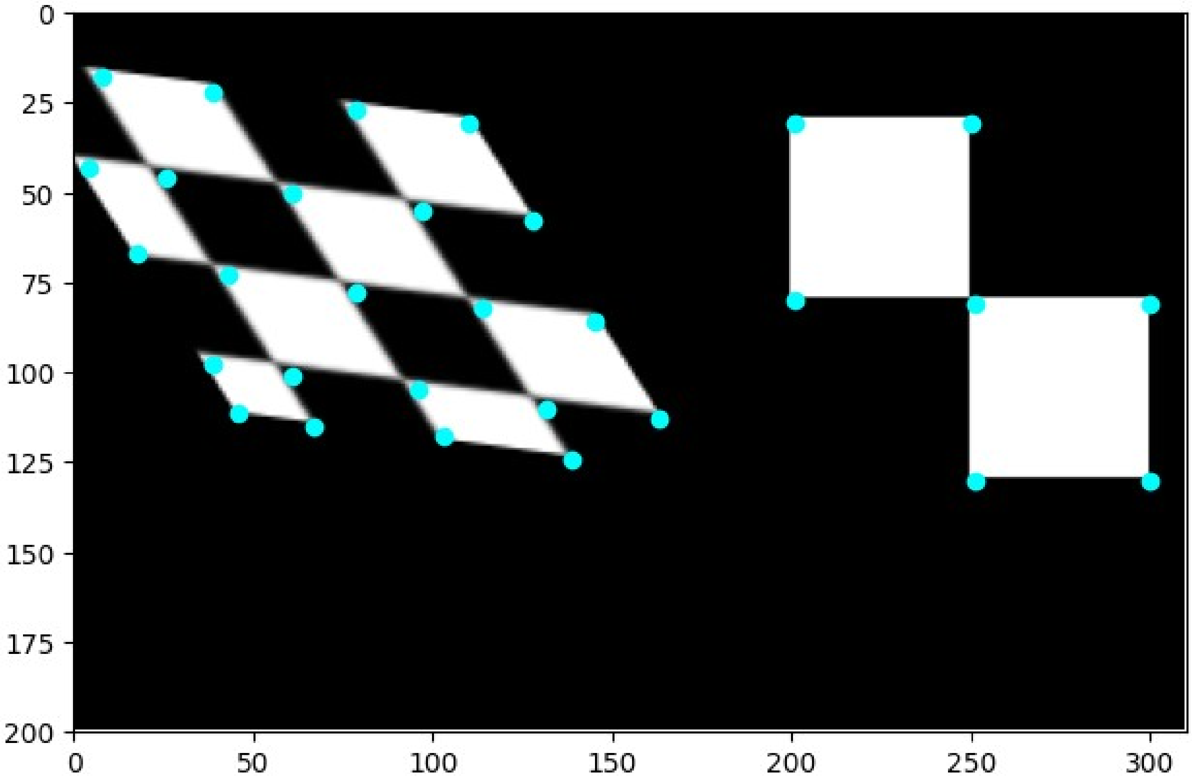
```
# Write your code in this cell.

def non_max_suppression(harris_response: np.array, distance: float,
threshold: float):
    candidate_detections = np.argwhere(harris_response > threshold)
    sorted_detections = sort_detections(candidate_detections,
harris_response)
    maxima = local_max(sorted_detections, distance)
    return maxima
```

### Answer 3.10

Execute the cell below and copy the saved image on Overleaf for Question 3.10. The cell below runs non-maximum suppression on the Harris response map and then shows the locations of the corner detections in cyan. Duplicate corner detections should now be removed.

```
coords = non_max_suppression(harris_response(image, 0.05, 3), 10.0,
0.02)
fig, ax = plt.subplots()
ax.imshow(image, cmap=plt.cm.gray)
ax.plot(coords[:, 1], coords[:, 0], color='cyan', marker='o',
        linestyle='None', markersize=6)
ax.axis((0, 310, 200, 0))
fig.tight_layout()
plt.show()
fig.savefig('Data/Solutions/question_3_10.pdf', format='pdf',
bbox_inches='tight')
```


0
25
50
75
100
125
150
175
200
0
50
100
150
200
250
300