Durham University

Computer Science

Summative Assignment

Module code and title	COMP2271 Data Science
Academic year	2023-24
Coursework title	Image Processing coursework
Coursework credits	5 credits
% of module's final mark	25%
Lecturer	Amir Atapour-Abarghouei
Submission date*	Tuesday, April 30, 2024 14:00
Estimated hours of work	10 hours
Submission method	Gradescope (code)

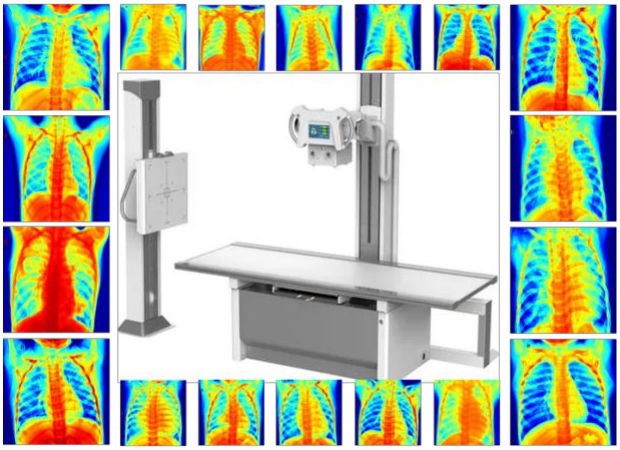
Additional coursework files	*classifier.model, classify.py, xray_images (directory containing 100 jpg files)*
Required submission items and formats	*Report (pdf), code (.py files), Results (directory containing 100 jpg files)*

* This is the deadline for all submissions except where an approved extension is in place.

Late submissions received within 5 working days of the deadline will be capped at 40%.
Late submissions received later than 5 days after the deadline will receive a mark of 0.
It is your responsibility to check that your submission has uploaded successfully and obtain a submission receipt.

Your work must be done by yourself (or your group, if there is an assigned groupwork component) and comply with the university rules about plagiarism and collusion. Students suspected of plagiarism, either of published or unpublished sources, including the work of other students, or of collusion will be dealt with according to University guidelines (https://www.dur.ac.uk/learningandteaching.handbook/6/2/4/).

Future Automated Medical Diagnosis via Image Processing



1. Background

Autonomous knowledge extraction pipelines, decision support mechanisms, and advanced medical diagnosis assistance systems are fast becoming a reality in day-to-day healthcare services. This is especially noticeable within the realm of medical image analysis as medical images are one of the primary data modalities contributing to automated digital health tools.

In this assignment, we are dealing with a dataset of 100 X-Ray images in JPEG format. Half of the images in the dataset are from healthy individuals and half are from patients who suffer from pneumonia. All chest X-ray imaging of this data was performed as part of patients' routine clinical care. Typically, normal chest X-ray images from healthy individuals present lungs without any areas of abnormal opacification in the image. However, patients with pneumonia typically exhibit white spots in the lungs that identify an infection. While the data you will be working with comes from the medical domain, note that **you do not need any specific domain knowledge to complete this assignment**.

You will take up the role of a data scientist with expertise in image processing who is expected to prepare the data for downstream applications, just as actual data scientists do on a daily basis. The data used in this assignment are colour images with a resolution of 256×256 pixels. This nature of the data might seem alien to a data scientist, such as you and me, but the preparation and initial analysis of data does not require in-depth familiarity with the subject area and is a routine part of the job of any data scientist or image processing expert. So, do not feel intimidated by the medical nature of this exercise.

2. The Task

In this scenario, you have been hired by a radiology clinic to perform some image processing. The clinic has provided you with some files to get you started in "**image_processing_files**". The dataset for this assignment contain 100 images in the directory:

“image_processing_files /xray_images/”

Within **image_processing_files**, you also have access to a python script **“classify.py”** which uses the model weights **“classifier.model”** that is used, in part, to evaluate your results.

The 256×256 images are test images which the radiology clinic that has hired you uses to evaluate the performance of their autonomous decision support systems. In this scenario, the clinic has created an AI system which contains an image classifier that receives as its input chest X-ray images and predicts as its output a binary label, which identifies the images as healthy or with pneumonia.

The classifier is already pre-trained, frozen and cannot change. **You are <u>not</u> expected to design or train a model or perform any machine learning of any kind.** Your job is to process the image data that is to be passed to this image classifier.

The chest X-ray images, however, suffer from a variety of issues caused by a broken scanner in the clinic (not really, the images have actually been corrupted artificially). These include:

- **Noise:** the images contain significant amounts of gaussian and salt/pepper noise.
- **Warping:** the images are distorted. This issue can potentially be resolved using projective transformations, so that objects within the images look as they should.
- **Contrast/brightness:** the contrast and brightness of the images are not adjusted and the details in these images are not visually clear.
- **Colour channel imbalance:** the information contained within the colour channels of the images is not balanced - i.e., some channels might be darker/brighter than others.
- **Missing region:** a circular portion of the image at the top right of all images is missing. This can be filled in using various inpainting methods.

Your task is to enhance the quality of these images using image processing techniques. Your results will be judged based on two factors:

1) **visual quality of the images**, as seen by a human observer (i.e., me!).

2) **performance of the pre-trained classifier** using your result images. The classifier is now basically guessing (a roughly 55% performance) on the images as they are now. Performance of 95% was achieved in my tests. There is no reason why 100% should not be possible. The script that runs the classifier is provided in **“classify.py”**, which uses the model weights **“classifier.model”**.

3. Hints

Here are a few hints to help get you started:

- Please read the submission instructions carefully. Since part of the marking will be automated, deviations from the instructions might lead to you losing marks as a result of simple submission mistakes.

- You are not expected or allowed to use any machine learning or deep learning. The assignment is meant to test your abilities in using conventional image processing techniques, though you can be as creative as you want in designing your solution.
- You will notice that during the warping process, only the **perspective** was affected.
- You can see that **multiple types of noise** are added to the image at the same time (Gaussian, Salt and Pepper).
- If you carefully inspect the images, you will notice that both their **contrast** and their **brightness** have been affected by the corruptions.
- You will notice that the missing region is not always in the same spot on the images, so the mask that has created this "hole" will not be constant for all images and has to be detected for each image. Your algorithm should generalise to unseen images as well so you can't just manually create a mask for every image in the dataset.
- You can complete the missing region of the images using various inpainting techniques. These include simple filling techniques such as filing pixel values with the average of their neighbours, using image inpainting techniques already built into OpenCV: https://docs.opencv.org/3.4.18/df/d3d/tutorial_py_inpainting.html

  or any number of more advanced techniques, such as:
  https://www.irisa.fr/vista/Papers/2004_ip_criminisi.pdf
  Make sure you carefully explain the inpainting method you use in your report.
- Though using the techniques covered in the lectures will get the task done, you are not limited to what was already covered. You will even get extra credit for finding more advanced, obscure and state-of-the-art techniques or proposing new processing methods that perform the required tasks.

4. Code Specifications

- Your program must operate with **OpenCV 4.1.x** on the lab PCs. You are not allowed to use any software packages that are not available on the lab PCs in Durham University. If you decide to create and operate your program on your own system, you will be responsible for handling the environment, required packages, dependencies and the required versions. All the software has been tested on my personal machine with Python 3.9.13, opencv-contrib-python 4.8.0.76 and opencv-python 4.8.0.76. As mentioned, the provided script has also been tested and can be run on lab PCs. Your code will be tested on the Lab PCs.
- Your program must contain an argument parser in the main script that allows a directory containing images to be specified. **I, as the tester, should be able to run** your program on the X-ray images in the following way:
  python main.py image_processing_files/xray_images/
  from which it will cycle through the images in the specified directory, perform all the processing and save the images **without changing the filenames** in a directory called "**Results**". The contents of this directory will also be a part of your submission. You program should create the **Results** directory if it does not already exist.

- Do not submit a Jupyter Notebook. Your final submission should only contain **.py** files. You may have multiple files, but the final program should be run through **main.py.**
- Your program should save the images in the same resolution as originally provided.
- You are not allowed to use any other image classification methods other than the pre-trained classifier provided for this assignment to measure the performance of your work. Only this script will be used during the marking.
- Your program will also be used to transform a set of 5 unseen images (intentionally withheld from you) as well as the images in "**xray_images**" directory provided. Note that your code will be tested on the unseen images using this command exactly: **python3 main.py unseen_test_imgs**

  I will of course provide the directory containing these unseen test images.
- Make sure you do not create any extra directories or folder structures. Your submission should contain all the files that you need as part of your source program with the only other directory submitted being the **Results** directory. Do not include additional directories. Your **main.py** and the **Results** directory should be in the **root**.
- Please do not include files that are not asked for. Do not include **classify.py**, as this will interfere with the evaluation script. The marking process will have its own version of this script. Please do not include the original files in **image_processing_files.** You only need to submit your result images, source code developed by you and your report.
- As mentioned before, your program must contain an argument parser in the main script that allows any directory containing images to be specified. This argument parser will be used to test your code. Do not hardcode image paths that might not exist when I run your code.

4. Submission

- Full program source code together with any required additional files for your final solution to the above task as a working python script, meeting the above *"**code specifications**"* for testing. Include all supporting python files and clear description of the code (*e.g., in a README.txt)*.
- A directory called "**Results**" which contains the results of your image enhancement techniques on the corrupted test images. Do **not** change the name, format or the resolution or the images. Note that the name of the directory "**Results**" should start with a capital R. Changes in the names and formats can affect the automatic grading system and will result in you losing marks.
- Report (max. 750 words) detailing your approach to the problem and the success of your solution in the tasks specified. Provide any illustrative images (as many as you feel necessary) of the intermediate results of the system you produce *(overlays, results of processing stages, etc.). **Any images, titles, captions, tables, references, and graphs do not count towards the total word count of the report.*** Summarise the success of your system in enhancing the quality of images and the effects of your image processing techniques on improving the performance of the downstream classifier on the test data set. Submit a PDF (not in any other format).

4. Marks

The marks for this assignment will be awarded as follows:

- Visual quality of the submitted images 20%
  ask yourself questions like these before submitting your results:
  - Has the noise been removed from the images?
  - Is the missing region "plausibly and realistically" filled?
  - Have the images been successfully dewarped?
  - Are details in the images visible?
  - Are the images blurry?
  - Don't limit yourself to these questions though, ask more!
- Performance of the image classifier on the submitted images 20%
- Clear and well documented code that works on unseen test images.
  Make sure you code follows the PEP8 coding style 10%
- Report:
  - Discussion/detail of solution design and choices made 15%
  - *Qualitative* / *quantitative* evidence and analysis of performance 15%
- Additional credit will be given for one or more of the following:
  - design and use of an alternative or novel methods
  - use of heuristics or advanced processing to improve performance
  - significant improvements in the performance of the classifier
  - novelty or quality of output for the inpainting technique used
    (for any of the above, up to a maximum, dependent on quality) 20%

Total: 100%

Plagiarism: You must not plagiarise your work. Attempts to hide plagiarism by simply changing comments/variable names will be detected. You should have been made aware of the Durham University policy on plagiarism.

Submissions should be made through the Gradescope system. Managing the submission is your responsibility. I highly recommend you do not leave your submission to the last minute. Due to high traffic right before the deadline, any system like this can get slow or crash and it is best to avoid issues like by giving yourself plenty of time to submit your assignment.

Gradescope handles directory submissions and zip files in its own way so you may have to make adjustments to your files and resubmit a few times to get things right. Leave some time for that as well. Do not submit any extra files that you are not asked to, in the assignment.

Automated tests on Gradescope will notify you if there are obvious issues, e.g., your submission is in the wrong format or if there are files missing. Ignoring those warnings will lead to a loss of marks. Give yourself plenty of time to be able to readjust your submission to avoid such issues.

Note that delayed submissions are handled by the department centrally and I do not have the power to grant extensions or disregard delays.