COLORADO STATE UNIVERSITY GLOBAL

CSC515: FOUNDATIONS OF COMPUTER VISION

Credit Hours: 3

Contact Hours: This is a 3-credit course, offered in accelerated format. This means that 16 weeks of material is covered in 8 weeks. The exact number of hours per week that you can expect to spend on each course will vary based upon the weekly coursework, as well as your study style and preferences. You should plan to spend 14-20 hours per week in each course reading material, interacting on the discussion boards, writing papers, completing projects, and doing research.

Faculty Information: Faculty contact information and office hours can be found on the faculty profile page.

COURSE DESCRIPTION AND OUTCOMES

Course Description:

In this Graduate course, students will apply digital image construction and processing. Students will explore topics associated with image formation, image acquisition, and image geometry. Students will be exposed to the techniques required to efficiently analyze images for representation in applicable context scenarios. Students will also apply image processing techniques for filtering and edge detection for image deconstruction.

Course Overview:

This course will introduce you to the foundations of computer vision. You will learn all the techniques necessary to process images for greater machine learning and neural network model efficiency and accuracy. This course will use both theory and practical application of the information learned.

Course Learning Outcomes:

1. Describe the mathematical tools used in digital image processing and computer vision.
2. Examine concepts, models and methods in the field of computer vision.
3. Compare multi-scale representation methods in computer vision.
4. Apply edge detection techniques to a given image.
5. Evaluate image filtering techniques for solving a given problem.
6. Analyze the accuracy of several problem-solving methods.

PARTICIPATION & ATTENDANCE

Prompt and consistent attendance in your online courses is essential for your success at CSU-Global Campus. Failure to verify your attendance within the first 7 days of this course may result in your withdrawal. If for some reason you would like to drop a course, please contact your advisor.

Online classes have deadlines, assignments, and participation requirements just like on-campus classes. Budget your time carefully and keep an open line of communication with your instructor. If you are having technical problems, problems with your assignments, or other problems that are impeding your progress, let your instructor know as soon as possible.

Course Materials

Required:

Bhuyan, M. K. (2019). *Computer vision and image processing: Fundamentals and applications*. Boca Raton, FL: CRC Press.
ISBN 9780815370840

***NOTE:** All non-textbook required readings and materials necessary to complete assignments, discussions, and/or supplemental or required exercises are provided within the course itself. Please read through each course module carefully.*

Course Schedule

Due Dates

The Academic Week at CSU-Global begins on Monday and ends the following Sunday.

- **Discussion Boards:** The original post must be completed by Thursday at 11:59 p.m. MT and Peer Responses posted by Sunday 11:59 p.m. MT. Late posts may not be awarded points.
- **Critical Thinking:** Assignments are due Sunday at 11:59 p.m. MT.

Weekly Reading and Assignment Details

Module 1

Readings

- Chapter 1 Section 1.1 in *Computer vision and image processing: Fundamentals and applications.*
- Davies, E.R. (2018). Vision, the challenge. In Computer vision (5th Edition). London Wall, London: Academic Press.
- Introduction to OpenCV-Python tutorials. (n.d.).
- Ravindran, P. (2015). Create superlative images with OpenCV. Electronics For You, 4(5).

Introduction Discussion

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Comparison of Computer Vision and Human Perception

Computer vision and human perception are two realizations of the process of seeing. One is done by machines and the other is done by people. But are these two actions similar only in the metaphorical sense, and therefore, should be kept separate? That is, are some things better analyzed via human perception and should only be done by humans, and vice versa? Or is there real substance to the metaphor, so that each could benefit from interacting with the other?

Address these questions in essay form. In your paper, be sure to discuss the similarities and differences between human perception and computer vision. Your paper should describe at least two reasons to keep the process of seeing in humans and machines separate, and two reasons supporting the benefits of their interaction. Substantiate your claims using at least six references (four peer reviewed) and cite them in your paper using correct APA styling.

Your paper should be 4-5 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least six scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: The Three Basic Categories of Computer Vision and Real World Applications

Computer vision is concerned with modeling and replicating human vision using computers and is divided into three basic categories as follows:

1. **Low-level vision** are the standard procedures to improve image quality and format. This includes standard image processing procedures. No "intelligence" is involved here.
2. **Mid-level vision** includes extracting features (like edges and corners) and characterizing components (like regions). This involves "some intelligence."
3. **High-level vision** includes recognizing patterns and comparing to known models. This is image understanding and involves "high intelligence" (i.e., machine learning).

In essay form, discuss some computer vision problems that can be solved using these categories. Your paper should describe at least two problems that can be solved for each of the three levels of computer vision. The details should include the problem and how the category might facilitate its solution. In addition, be sure to explain how all three levels can work together for high accuracy solutions. Substantiate your claims using at least six references (four peer reviewed) and cite them in your paper using correct APA styling.

Your paper should be 4-5 pages in length and conform to *CSU Global Guide to Writing and APA*. Include at least six scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Portfolio Milestone (50 points)

OPTION #1 Installing OpenCV 1

It is time to begin thinking about your Portfolio Project. In order to complete the Portfolio Project, OpenCV will need to be installed and working properly on your desktop.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.

For this milestone assignment, install OpenCV based on your specific operating system. Then, use OpenCV to complete the following:

1. Write Python code to import this image.
2. Write Python code to display the image.
3. Write Python code to write a copy of the image to any directory on your desktop.

Your submission should be one executable Python file.

OPTION #2 Installing OpenCV 2
It is time to begin thinking about your Portfolio Project. In order to complete the Portfolio Project, OpenCV will need to be installed and working properly on your desktop.

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in commercial products.

For this milestone assignment, install OpenCV based on your specific operating system. Then, use OpenCV to complete the following:

1. Write Python code to import this image.
2. Write Python code to display the image.
3. Write Python code to write a copy of the image to any directory on your desktop.

Your submission should be one executable Python file.

Module 2

Readings

- Chapter 1 Sections 1.2 and 1.3 in *Computer vision and image processing: Fundamentals and applications*
- Arithmetic operations on images. (n.d.). Retrieved from https://docs.opencv.org/master/d0/d86/tutorial_py_image_arithmetics.html
- Basic operations on images. (n.d.). Retrieved from https://docs.opencv.org/master/d3/df2/tutorial_py_basic_ops.html

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Puppy Image Multi-scale Representation in OpenCV

After completing the Required Reading, you should have a good idea of how to use OpenCV for multi-scale representation of images by pixels matrices. Take a look at this image of a puppy. Being a colored image, this image has three
channels, corresponding to the primary colors of red, green, and blue.

1. Import this image (using the link) into OpenCV and write code to extract each of these channels separately to create 2D images. This means that from the n x n x 3 shaped image, you will get 3 matrices of the shape n x n.
2. Now, write code to merge all these images back into a colored 3D image.
3. What will the image look like if you exchange the reds with the greens? Write code to merge the 2D images created in step 1 back together, this time swapping out the red channel with the green channel (GRB).

Be sure to display the resulting images for each step. Your submission should be one executable Python file.

OPTION #2: Kitten Image Multi-scale Representation in OpenCV

After completing the Required Reading, you should have a good idea of how to use OpenCV for multi-scale representation of images by pixels matrices. Take a look at this image of a kitten. Being a colored image, this image has three
channels, corresponding to the primary colors of red, green, and blue.

1. Import this image (using the link) into OpenCV and write code to extract each of these channels separately to create 2D images. This means that from the n x n x 3 shaped image, you'll get 3 matrices of the shape n x n.
2. Now, write code to merge all these images back into a colored 3D image.
3. What will the image look like if you exchange the reds with the greens? Write code to merge the 2D images created in step 1 back together, this time swapping out the blue channel with the red channel (GRB).

Be sure to display the resulting images for each step. Your submission should be one executable Python file.

Module 3

Readings

- Chapter 2 Sections 2.1 and 2.2 in *Computer vision and image processing: Fundamentals and applications*
- Changing colorspaces. (n.d.). Retrieved from https://docs.opencv.org/master/df/d9d/tutorial_py_colorspaces.html
- Geometric transformations of images. (n.d.). Retrieved from https://docs.opencv.org/master/da/d6e/tutorial_py_geometric_transformations.html
- Histogram equalization. (n.d.). Retrieved from https://docs.opencv.org/master/d5/daf/tutorial_py_histogram_equalization.html
- Yip, B., Towner, R., Kling, T., Chen, C., & Wang, Y. (2018). Image pre-processing using OpenCV Library on MORPH-II Face Database.

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Simple Image Processing in OpenCV 1

A facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. In general, facial recognition systems work by comparing selected facial features from a given image with faces within a database. Good facial recognition systems take into account the variation of pose, illumination of the image and facial expression. They also eliminate backgrounds and hairstyles, which are image properties that are not
useful for computer vision tasks.

Recall that the MORPH-II dataset includes 55,134 mugshots with longitudinal spans taken between 2003 and late 2007. Write algorithms to process both images for subject 1 and subject 2. Processed images for each subject should be ideal for facial recognition accuracy. Final images for each subject should be in grayscale, adjusted for the effect on illumination, rotated based on the angle between the eye centers, cropped and scaled to create a new bounding box for the face. All final image bounding boxes should be the same dimensions.

Be sure to display the final processed images for each subject. Your submission should be an executable Python file.

OPTION #2: Simple Image Processing in OpenCV 2

A facial recognition system is a technology capable of identifying or verifying a person from a digital image or a video frame from a video source. In general, facial recognition systems work by comparing selected facial features from a given image with faces within a database. Good facial recognition systems take into account the variation of pose, illumination of the image, and facial expression. They also eliminate backgrounds and hairstyles, which are image properties that are not
useful for computer vision tasks.

Recall that the MORPH-II dataset includes 55,134 mugshots with longitudinal spans taken between 2003 and late 2007. Write algorithms to process both images for subject 1 and subject 2. Processed images for each subject should be ideal for facial recognition accuracy. Final images for each subject should be in grayscale, adjusted for the effect on illumination, rotated based on the angle between the eye centers, cropped and scaled to create a new bounding box for the face. All final image bounding boxes should be the same dimensions.

Be sure to display the final processed images for each subject. Your submission should be an executable Python file.

Portfolio Milestone (50 points)

OPTION #1: Drawing Functions in OpenCV

It is time to think more about your upcoming Portfolio Project. In face and object detection, it is often useful to draw on the image. Perhaps you would like to put bounding boxes around features or use text to tag objects/people in the image.

Use a camera to take a picture of yourself facing the frontal. In OpenCV, draw on the image a red bounding box for your eyes and a green circle around your face. Then tag the image with the text “this is me”.

Your submission should be one executable Python file.

OPTION #2: Blending Images in OpenCV

It is time to think more about your upcoming Portfolio Project. In face and object detection, it is often useful to create a new image by combining two images together. Perhaps you would like to remove a tattoo or create a deepfake.

Use a camera to take a picture of yourself facing the frontal. Using an image from the web, create a deepfake by replacing someone else’s likeness with yourself.

Your submission should be one executable Python file.

Module 4

Readings

- Chapter 2 Sections 2.3 and 2.4 in *Computer vision and image processing: Fundamentals and applications*
- Davies, E.R. (2018). Image filtering and morphology. In *Computer vision*. London Wall, London: Academic Press.
- Fourier transform. (n.d.). Retrieved from https://docs.opencv.org/master/de/dbc/tutorial_py_fourier_transform.html
- Image smoothing. (n.d.). Retrieved from https://docs.opencv.org/master/d4/d13/tutorial_py_filtering.html

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Mean, Median, and Gaussian Filters

Image filtering involves the application of window operations that perform useful functions, such as noise removal and image enhancement. Compare the effects of mean, median, and Gaussian filters on an image for different kernel windows.

This image contains impulse noise. In OpenCV, write algorithms for this image to do the following:

1. Apply mean, median, and Gaussian filters using a 3x3 kernel. Additionally, for Gaussian, select two different values of sigma. Think about how to select good values of sigma for optimal results.
2. Apply mean, median, and Gaussian filters using a 5x5 kernel. For Gaussian, use the same values of sigma you selected in the above step.
3. Apply mean, median, and Gaussian filters using a 7x7 kernel. For Gaussian, use the same values of sigma you selected in the above step.

Output your filter results as 3 x 4 side-by-side subplots to make comparisons easy to inspect visually. That is, your subplot should have 3 rows (1 for each kernel size) and 4 columns (1 for each filter type, 2 for Gaussian). Be sure to include row and column labels.

Next, write a 2-3 page summary of your output results. Include in your summary, the following:

1. Which filter type is preferred for removal of impulse noise and why? Provide two references of support for your answer and cite them in your summary using correct APA styling.
2. Which filter (include kernel size and sigma, if applicable) performed the best visually? Include details like whether there were image features better preserved and/or better enhanced. Are these preservations and enhancements of image features important? Why or why not?
3. Are your results in line with the preferred method? Discuss why or why not?

Your submission should be one executable Python script and one summary of 2-3 pages in length that conforms to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Laplacian Filters for Different Kernel Windows

Image filtering involves the application of window operations that perform useful functions, such as noise removal and image enhancement. Compare the effects of Laplacian and Gaussian filters on an image for different kernel windows.

This image contains Gaussian noise. In OpenCV, write algorithms for this image to do the following:

1. Apply a Gaussian, a Laplacian, and a Gaussian with Laplacian filter using a 3x3 kernel. For Gaussian, think about how to select a good value of sigma for optimal results.
2. Apply a Gaussian, a Laplacian, and a Gaussian with Laplacian filter using a 5x5 kernel. For Gaussian, use the same value of sigma you selected in the above step.
3. Apply a Gaussian, a Laplacian, and a Gaussian with Laplacian filter using a 7x7 kernel. For Gaussian, use the same value of sigma you selected in the above step.

Output your filter results as 3 x 3 side-by-side subplots to make comparisons easy to inspect visually. That is, your subplot should have 3 rows (1 for each kernel size) and 3 columns (1 for each filter type). Be sure to include row and column labels.

Next, write a 2-3 page summary of your output results. Include in your summary, the following:

1. Which filter type (Gaussian, Laplacian, or Gaussian with Laplacian) is preferred for better edge detection and why? Provide two references of support for your answer and cite them in your summary using correct APA styling.
2. Which filter (include kernel size and sigma, if applicable) performed the best visually? Include details like whether there were image features better preserved and/or better enhanced. Are these preservations and enhancements of image features important for edge detection? Why or why not?
3. Are your results in line with the preferred method? Discuss why or why not?

Your submission should be one executable Python script and one summary of 2-3 pages in length that conforms to *CSU Global Guide to Writing and APA*. Include at least two scholarly references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Module 5

Readings

- Chapter 2 Section 2.5 in *Computer vision and image processing: Fundamentals and applications*
- Davies, E.R. (2018). Image filtering and morphology. In *Computer vision*. London Wall, London: Academic Press.
- Morphological transformations. (n.d.). Retrieved from https://docs.opencv.org/master/d9/d61/tutorial_py_morphological_ops.html

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Morphology Operations for Fingerprint Enhancement

Fingerprints are tiny patterns on the tip of every finger. The uniqueness of fingerprints has been studied and it is well established that the probability of two fingerprints matching is extremely small. Since they are unique, fingerprints can be used to identify or confirm the identity of an individual.

Fingerprint recognition refers to the automated method of identifying or confirming the identity of an individual based on the comparison of two fingerprints. These comparisons can be made based on the type and the position of the ridge characteristics. The performance of fingerprint recognition applications heavily rely on the input fingerprint image quality.

A latent fingerprint is left on a surface by deposits of oils and/or perspiration from the finger. It is not usually visible but may be detected with special techniques such as dusting for fingerprints. In order to reduce rejection rates in most cases the acquired latent fingerprints have to be enhanced prior to matching to reduce the degradation, noise, or incompleteness. Enhancement can be achieved using morphological image processing.

Acquire an image of a latent fingerprint. In OpenCV, write algorithms to process the image using morphological operations (dilation, erosion, opening, and closing).

Next, write a 2-3 page summary of your observed results. Include in your summary, the following:

1. Describe in detail what enhancements did each morphological operation make on the image and how beneficial these enhancements are for fingerprint recognition.
2. Did the enhancement also result in data loss of other features? Explain.
3. Research morphological operations for fingerprint enhancements and include whether your results were similar with the findings in these. Be sure to cite them in your summary using correct APA styling.

Your submission should be one executable Python script and one summary of 2-3 pages in length that conforms to *CSU Global Guide to Writing and APA*. Include at least two references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Morphology Operations for Handwritten Text Enhancement

Many companies have thousands of documents to process, analyze, and transform in order to carry out day-to-day operations. Some of these documents might contain handwritten text. Handwritten text can be found in handwritten notes, memos, whiteboards, medical records, historical documents, text input by stylus, etc. Therefore, a complete Optical Character Recognition (OCR) solution has to include support for recognizing handwritten text in images.

Handwriting recognition (HWR), also known as Handwritten Text Recognition (HTR), is the ability of a computer to receive and interpret intelligible handwritten input. A handwriting recognition system handles formatting, performs correct segmentation into characters, and finds the most plausible words. Due to the variety of handwriting styles, orientations, character sizes, and distortions, handwriting recognition is a challenging problem. Thus, to reduce the rejection rates during the matching stage, handwritten text has to be enhanced prior to matching. Enhancement can be achieved using morphological image processing.

Acquire a scanned input image of cursive handwritten text on a sticky note. In OpenCV, write algorithms to process the image using morphological operations (dilation, erosion, opening, and closing).

Next, write a 2-3 page summary of your observed results. Include in your summary, the following:

1. Describe in detail what enhancements did each morphological operation make on the image and how beneficial these enhancements are for handwriting recognition.
2. Did the enhancement also result in data loss of other features? Explain.
3. Research morphological operations for handwriting enhancements and include whether your results were similar with the findings in these. Be sure to cite them in your summary using correct APA styling.

Your submission should be one executable Python script and one summary of 2-3 pages in length that conforms to *CSU Global Guide to Writing and APA*. Include at least two references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Module 6

Readings

- Chapter 2 Section 2.6 in *Computer vision and image processing: Fundamentals and applications*
- Davies, E.R. (2018). The role of thresholding. In *Computer vision*. London Wall, London: Academic Press.
- Image segmentation with watershed algorithm. (n.d.). Retrieved from https://docs.opencv.org/master/d3/db4/tutorial_py_watershed.html
- Image thresholding. (n.d.). Retrieved from https://docs.opencv.org/master/d7/d4d/tutorial_py_thresholding.html

Discussion (25 points)

Critical Thinking (75 points)

OPTION #1: Adaptive Thresholding Scheme for Simple Objects
If an image has been preprocessed properly to remove noise, a key step that is generally used when interpreting that image is segmentation. Image segmentation is a process in which regions or features sharing similar characteristics are identified and grouped together.

The thresholds in the algorithms discussed in this module were chosen by the designer. In order to make segmentation stronger to variations in the scene, the algorithm should be able to select an appropriate threshold automatically using the amount of intensity present in the image. The knowledge about the gray values of objects should not be hard-coded into an algorithm. The algorithm should use knowledge about the relative characteristics of gray values to select the appropriate threshold. A thresholding scheme that uses such knowledge and selects a proper threshold value for each image without human intervention is called an adaptive thresholding scheme.

Find on the internet (or use a camera to take) three different types of images: an indoor scene,outdoor scenery, and a close-up scene of a single object. Implement an adaptive thresholding scheme to segment the images as best as you can.

Your submission should be one executable Python script and one summary of 2-3 pages in length that conforms to *CSU Global Guide to Writing and APA*. Include at least two references in addition to the course textbook. The CSU Global Library is a good place to find these references.

OPTION #2: Adaptive Thresholding Scheme Accounting for Changes in Illumination

If an image has been preprocessed properly to remove noise a key step that is generally used when interpreting that image is segmentation. Image segmentation is a process in which regions or features sharing similar characteristics are identified and grouped together.

The thresholds in the algorithms discussed in this module were chosen by the designer. In order to make segmentation stronger to variations in the scene, the algorithm should be able to select an appropriate threshold automatically using the amount of intensity present in the image. The knowledge about the gray values of objects should not be hard-coded into an algorithm. The algorithm should use knowledge about the relative characteristics of gray values to select the appropriate threshold. A thresholding scheme that uses such knowledge and selects a proper threshold value for each image without human intervention is called an adaptive thresholding scheme.

Select two objects, one light and one dark. Put both objects on a flat surface. Illuminate the scene so that light comes from the direction of the light object. Take a picture of the scene. Develop a suitable adaptive thresholding scheme to segment the image as best as you can. Now, change the direction of illumination by moving your light source so that the illumination is from the direction of the dark object. Again, take a picture of the scene. Using this image, see whether your algorithm still works. If not, revise the algorithm to make it work.

Your submission should be one executable Python script and one summary of 2-3 pages in length that conforms to *CSU Global Guide to Writing and APA*. Include at least two references in addition to the course textbook. The CSU Global Library is a good place to find these references.

Module 7

Readings

- Chapter 3 Section 3.2 in *Computer vision and image processing: Fundamentals and applications*
- Canny edge detection. (n.d.). Retrieved from https://docs.opencv.org/master/da/d22/tutorial_py_canny.html
- Davies, E.R. (2018). Edge detection. In *Computer vision*. London Wall, London: Academic Press.
- Image gradients. (n.d.). Retrieved from https://docs.opencv.org/master/d5/d0f/tutorial_py_gradients.html

Discussion (25 points)

Module 8

Readings

- Chapter 4 Section 4.1 in *Computer vision and image processing: Fundamental Pattern Recognition Concepts*
- Cascade classifier. (n.d.). Retrieved from
- https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html
- Davies, E.R. (2018). Face detection and recognition. In *Computer vision*. London Wall, London: Academic Press.
- Davies, E.R. (2018). Surveillance. In *Computer vision*. London Wall, London: Academic Press.
    - Read section 22.10 Licence plate location
- Jalled, F. & Voronkov, I. (2016). Object detection using image processing.

Discussion (25 points)

Portfolio Project (250 points)

OPTION #1: License Plate Detection and Faked Content

Reliable recognition of objects is a challenging task for artificial intelligence, yet humans are able to perform these kinds of tasks with ease. In face and object detection, it is often useful to create a new image by combining two images together. Perhaps you would like to remove a tattoo or create a deepfake.

The goal of this project is to write algorithms for license plate detection and faked content creation. Select three color images from the internet that meet the following requirements:

1. Two images containing vehicles with Russian license plates and one image of vehicles with a non-Russian plate.
2. All images should include the entire vehicle and not just the license plate.
3. At least one image with Russian plates should display the license plate far away.
4. At least one image should include multiple vehicles.

5. All images should vary in light illumination and color intensity.

First, using the appropriate trained cascade classifier, write one algorithm to detect the Russian license plate in the gray scaled versions of the original images. Put a red boundary box around the detected plate in the image in order to see what region the classifier deemed as a license plate. If expected results are not achieved on the unprocessed images, apply processing steps before implementing the classifier for optimal results.

After the license plates have been successfully detected, you will want to use them to create faked content by swapping out the detected license plate in one of them with another. It is important that the swapped license plates are properly rotated and scaled so that your fakes are believable.

Inspect your results and write a summary describing the techniques you used to detect and swap out the Russian license plates in your images. Reflect on the challenges you faced and how you overcame these challenges. Furthermore, discuss in your summary, the accuracy of your results for all three images and techniques you used to improve the accuracy after each repeated experiment.

Your submission should be one executable Python script and one summary of 2-3 pages in length.

OPTION #2: Face Detection and Privacy

To address privacy concerns you may want to use data anonymization. On images, this can be achieved by hiding features that could lead to a person or personal data identification, such as the person's facial features or a license plate number.

The goal of this project is to write algorithms for face detection and feature blurring. Select three color images from the web that meet the following requirements:

1. Two images containing human subjects facing primarily to the front and one image with a non-human subject.
2. At least one image of a human subject should contain that person's entire body.
3. At least one image should contain multiple human subjects.
4. At least one image should display a person's face far away.
5. All images should vary in light illumination and color intensity.

First, using the appropriate trained cascade classifier, write one algorithm to detect the human faces in the gray scaled versions of the original images. Put a red boundary box around the detected face in the image in order to see what region the classifier deemed as a human face. If expected results are not achieved on the unprocessed images, apply processing steps before implementing the classifier for optimal results.

After the faces have been successfully detected, you will want to process **only the extracted faces** before detecting and applying blurring to hide the eyes. Although the eye classifier is fairly accurate, it is important that all faces are centered, rotated, and scaled so that the eyes are perfectly aligned. If expected results are not achieved, implement more image processing for optimal eye recognition. Now, apply a blurring method to blur the eyes out in the extracted image.

Inspect your results and write a summary describing the techniques you used to detect and blur the eyes out of human faces in images. Reflect on the challenges you faced and how you overcame these challenges. Furthermore, discuss in your summary, the accuracy of your results for all three images and techniques you used to improve the accuracy after each repeated experiment.

Your submission should be one executable Python script and one summary of 2-3 pages in length.

COURSE POLICIES

Grading Scale	
A	95.0 – 100
A-	90.0 – 94.9
B+	86.7 – 89.9
B	83.3 – 86.6
B-	80.0 – 83.2
C+	75.0 – 79.9
C	70.0 – 74.9
D	60.0 – 69.9
F	59.9 or below

Course Grading

20% Discussion Participation
45% Critical Thinking Assignments
35% Final Portfolio Project
0% Live Classroom

In-Classroom Policies

For information on late work and incomplete grade policies, please refer to our **In-Classroom Student Policies and Guidelines** or the Academic Catalog for comprehensive documentation of CSU-Global institutional policies.

Academic Integrity

Students must assume responsibility for maintaining honesty in all work submitted for credit and in any other work designated by the instructor of the course. Academic dishonesty includes cheating, fabrication, facilitating academic dishonesty, plagiarism, reusing /re-purposing your own work (see *CSU-Global Guide to Writing and APA Requirements* for percentage of repurposed work that can be used in an assignment), unauthorized possession of academic materials, and unauthorized collaboration. The CSU-Global Library provides information on how students can avoid plagiarism by understanding what it is and how to use the Library and Internet resources.

Citing Sources with APA Style

All students are expected to follow the *CSU-Global Guide to Writing and APA Requirements* when citing in APA (based on the APA Style Manual, 6th edition) for all assignments. For details on CSU-Global APA style, please review the APA resources within the CSU-Global Library under the “APA Guide & Resources” link. A link to this document should also be provided within most assignment descriptions in your course.

Disability Services Statement

CSU–Global is committed to providing reasonable accommodations for all persons with disabilities. Any student with a documented disability requesting academic accommodations should contact the Disability Resource Coordinator at 720-279-0650 and/or email ada@CSUGlobal.edu for additional information to coordinate reasonable accommodations for students with documented disabilities.

Netiquette

Respect the diversity of opinions among the instructor and classmates and engage with them in a courteous, respectful, and professional manner. All posts and classroom communication must be conducted in accordance with the student code of conduct. Think before you push the Send button. Did you say just what you meant? How will the person on the other end read the words?

Maintain an environment free of harassment, stalking, threats, abuse, insults or humiliation toward the instructor and classmates. This includes, but is not limited to, demeaning written or oral comments of an ethnic, religious, age, disability, sexist (or sexual orientation), or racist nature; and the unwanted sexual advances or intimidations by email, or on discussion boards and other postings within or connected to the online classroom. If you have concerns about something that has been said, please let your instructor know.