**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **10/07/2020** | **Name:** | **Gaganashree P** |
| **Course:** | **Coursera** | **USN:** | **4AL15EC 024** |
| **Topic:** | **Digital image processing 4: Application** | **Semester & Section:** | **8th - A** |
| **GitHub Repository:** | **Gaganashree-P** |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
|  |

**REPORT –**

Introduction to image processing

Image processing is a method to perform some operations on an image, in order to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

* Importing the image via image acquisition tools;
* Analysing and manipulating the image;
* Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analogue and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data have to undergo while using digital technique are pre-processing, enhancement, and display, information extraction.

In this lecture we will talk about a few fundamental definitions such as image, digital image, and digital image processing. Different sources of digital images will be discussed and examples for each source will be provided. The continuum from image processing to computer vision will be covered in this lecture. Finally we will talk about image acquisition and different types of image sensors.

**Filtering an Image**

**Image filtering is useful for many applications, including smoothing, sharpening, removing noise, and edge detection. A filter is defined by a kernel, which is a small array applied to each pixel and its neighbors within an image. In most applications, the center of the kernel is aligned with the current pixel, and is a square with an odd number (3, 5, 7, etc.) of elements in each dimension. The process used to apply filters to an image is known as convolution, and may be applied in either the spatial or frequency domain. See**[**Overview of Transforming Between Image Domains**](https://northstar-www.dartmouth.edu/doc/idl/html_6.2/Overview_of_Transforming_Between_Image_Domains.html)**for more information on image domains.**

**Within the spatial domain, the first part of the convolution process multiplies the elements of the kernel by the matching pixel values when the kernel is centered over a pixel. The elements of the resulting array (which is the same size as the kernel) are averaged, and the original pixel value is replaced with this result. The CONVOL function performs this convolution process for an entire image.**

**Within the frequency domain, convolution can be performed by multiplying the FFT (Fast Fourier Transform) of the image by the FFT of the kernel, and then transforming back into the spatial domain. The kernel is padded with zero values to enlarge it to the same size as the image before the forward FFT is applied. These types of filters are usually specified within the frequency domain and do not need to be transformed. IDL's DIST and HANNING functions are examples of filters already transformed into the frequency domain. See**[**Windowing to Remove Noise**](https://northstar-www.dartmouth.edu/doc/idl/html_6.2/Removing_Noise.html#wp1030834)**for more information on these types of filters.**

**The following examples in this section will focus on some of the basic filters applied within the spatial domain using the CONVOL function:**

* **[Low Pass Filtering](https://northstar-www.dartmouth.edu/doc/idl/html_6.2/Filtering_an_Imagehvr.html" \l "wp1022750)**

* **[High Pass Filtering](https://northstar-www.dartmouth.edu/doc/idl/html_6.2/Filtering_an_Imagehvr.html" \l "wp1022814)**

* **[Directional Filtering](https://northstar-www.dartmouth.edu/doc/idl/html_6.2/Filtering_an_Imagehvr.html" \l "wp1027824)**

* **[Laplacian Filtering](https://northstar-www.dartmouth.edu/doc/idl/html_6.2/Filtering_an_Imagehvr.html" \l "wp1027822)**

**Since filters are the building blocks of many image processing methods, these examples merely show how to apply filters, as opposed to showing how a specific filter may be used to enhance a specific image or extract a specific shape. This basic introduction provides the information necessary to accomplish more advanced image-specific processing.**

**Low Pass Filtering**

**A low pass filter is the basis for most smoothing methods. An image is smoothed by decreasing the disparity between pixel values by averaging nearby pixels (see**[**Smoothing an Image**](https://northstar-www.dartmouth.edu/doc/idl/html_6.2/Smoothing_an_Image.html)**for more information).**

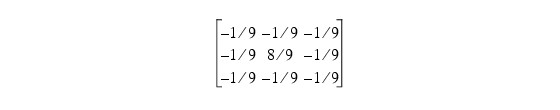
**Using a low pass filter tends to retain the low frequency information within an image while reducing the high frequency information. An example is an array of ones divided by the number of elements within the kernel, such as the following 3 by 3 kernel:**

****

### High Pass Filtering

A high pass filter is the basis for most sharpening methods. An image is sharpened when contrast is enhanced between adjoining areas with little variation in brightness or darkness (see [Sharpening an Image](https://northstar-www.dartmouth.edu/doc/idl/html_6.2/Sharpening_an_Image.html) for more detailed information).

A high pass filter tends to retain the high frequency information within an image while reducing the low frequency information. The kernel of the high pass filter is designed to increase the brightness of the center pixel relative to neighboring pixels. The kernel array usually contains a single positive value at its center, which is completely surrounded by negative values. The following array is an example of a 3 by 3 kernel for a high pass filter:



**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **10/07/2020** | **Name:** | **Gaganashree P** |
| **Course:** | **Salesforce** | **USN:** | **4AL15EC024** |
| **Topic:** | **Trailhead basic** | **Semester & Section:** | **8TH SEM &A Section** |
| **Github Repository:** | **Gaganashree-P** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS**   Welcome to Trailhead Every journey starts with a single step—and so does every trail. You’ve already taken the first step of your learning journey by opening Trailhead.  Trailhead is the fun way to learn. Whether you’re an admin just starting out, a graduate fresh from college, a Salesforce user, or someone who loves to learn, there’s something on Trailhead for you. How Does It Work? In Trailhead, learning topics are organized into modules, which are broken up into units. To finish a unit, you earn points by completing a quiz or a challenge. A quiz checks your knowledge with multiple-choice questions, while a challenge tests your skills by getting your hands dirty in a Salesforce org. Once you’ve finished all of the units in a module, you get a shiny new badge for your profile.  Trails are groups of modules that provide guided learning paths suited to specific roles or needs. You can also blaze your own trail, and choose the modules that you’re most interested in without following a pre-existing trail. Projects and superbadges allow you to learn interactively by requiring you to implement a feature or solution in an org.  You just learned a lot of Trailhead terms in just a couple of paragraphs. If it seems like a lot to keep straight, don’t worry. Let us be your trail guide, and work through the rest of this module to learn the ins and outs of Trailhead. Who Is Trailhead For? The short answer is, well, everyone. We have content for every role within an org, and every level of experience. We even have material for learning outside of Salesforce, such as modules on [Blockchain Basics](https://trailhead.salesforce.com/content/learn/modules/blockchain-basics" \t "_blank) and [European Union Privacy Law Basics](https://trailhead.salesforce.com/modules/european-union-privacy-law-basics), and a trail to help you [Get Started with iOS App Development](https://trailhead.salesforce.com/content/learn/trails/start-ios-appdev).  We have a variety of trails for a variety of learners, from representatives using Service Cloud to advanced Salesforce developers to admins at nonprofits who use Salesforce. And because we know that you're blazing trails all over the world, Trailhead content is available in six languages: English, German, Japanese, French, Spanish, and Portuguese. To see Trailhead in another language, scroll down to the footer of any page and choose a language from the dropdown menu.  Trailhead content blends a variety of content types and approaches to cater to all types of learners, including:   * Videos * Code samples * Walkthroughs * Screenshots * Images * Tables and charts * Step-by-step instructions   When we create content, we use a set of guiding principles that include:   * Practical examples and scenarios * Storytelling * Casual voice and tone * Positivity * Fun * Fun * Fun   And did we mention fun? We want you—and everyone who uses Trailhead—to have fun. We gamify our content so that you can learn new skills while you earn points and badges, and have something to show for it at the end. Where Do I Start? First things first: To complete challenges, earn badges, and keep track of your progress, you need to create an account. To create an account, click the **Sign Up** button from anywhere in Trailhead.  You have a few options. You can sign up for Trailhead with Salesforce, Google, LinkedIn, or your email address.  Choose whatever is easiest for you. We use your identity for login purposes only, and each option has its benefits. If you choose a social identity, you can keep your Trailhead profile no matter where your career takes you. If you choose your company Salesforce login, your company can use the Trail Tracker app to see all the awesome learning you're doing. If you want to attach both or modify your login identities after signup, you can do so anytime from the settings page on your profile.  To link your social accounts to an existing Trailhead account, click your profile image in the top-right corner of Trailhead and then click **Settings**. Scroll down and click **Connect** to connect your Salesforce, email, or social account, respectively. If the login identity you're trying to link already has Trailhead data, such as badges, points, or favorites, associated with it, you can merge it with your other account.  From this page, you can also manage your hands-on orgs (Trailhead Playgrounds or Developer Edition orgs) or link an existing org to your Trailhead account. If you don’t know what a Trailhead Playground is, read on. What's a Hands-on Challenge, Anyway? Learning is the bread and butter of Trailhead, and one of the best ways to learn is by doing. Earning a badge on Trailhead is more than just reading through modules. You also have to complete either a multiple-choice quiz or a hands-on challenge at the end of each unit. You’re probably familiar with multiple-choice quizzes (think Who Wants To Be A Millionaire?), but hands-on challenges are unique to Trailhead.  A hands-on challenge is more involved than a quiz and, as a result, earns you more points. To complete a hands-on challenge, you have to look at a set of requirements and do something in a Salesforce org to meet those requirements. What you have to do depends on what you’re learning. You could be writing an Apex class, creating a Lightning web component, or writing a field-level validation formula.  We even provide you with your own org, called a Trailhead Playground, that you can use to solve hands-on challenges. What’s a Trailhead Playground? When you encounter your first hands-on challenge, you’ll see a dropdown that lets you launch your hands-on org. A Trailhead Playground is an org that you can use for hands-on challenges, learning new features, and testing customizations. If you haven’t created one, don’t worry. You get one automatically when you create a Trailhead account  You can customize your Trailhead Playground however you want, and it comes with a set of Trailhead-specific data that you can use when completing challenges. Although Trailhead Playgrounds have limits (you can create only two users in a Trailhead Playground, for instance), they give you the same customization options as a production org. And a Trailhead Playground never expires, as long as you keep using it.  You can also use a Developer Edition org to complete hands-on challenges. Really, a Developer Edition org is the same thing as a Trailhead Playground, but Trailhead Playgrounds come with Trailhead-specific data, and a pre-installed unmanaged package that we use to test your hands-on challenge solutions. Unlike a Developer Edition org, you can spin up a Trailhead Playground with the push of a button, without filling out any forms, right from Trailhead, and then launch it with just a click when you need it. |