📝 Notas

11 sept 2025

## Visión por Computadora I (3/8) - a213b25

Invitados [mdorogov@fi.uba.ar](mailto:mdorogov@fi.uba.ar) [Visión por Computadora I - CEIA 21Co2025](mailto:vpc1_a21@cursoscapse.com)

Archivos adjuntos [Visión por Computadora I (3/8) - a213b25](https://www.google.com/calendar/event?eid=Yms2NW9obGNpN2M5c2lmNjdsdTFiYWlodGsgZ29vZ2xlLm1lZXRAbHNlLXBvc2dyYWRvcy5maS51YmEuYXI)

Registros de la reunión [Transcripción](?tab=t.u1jinfi2mh3d) [Grabación](https://drive.google.com/file/d/1SK6wKzQfkd2opvDHvVmqxZY-_BRSyvBt/view?usp=drive_web)

### Resumen

lse posgrados introdujo los filtros convolucionales, destacando que los histogramas pueden ser engañosos sin considerar la distribución espacial, y explicó que herramientas como la transformada de Fourier abordan esta limitación. La discusión de los participantes, incluyendo a Jonathan Borda, Daniel Bazan, Luciano Ceballos, Erlin Rey, Sebastian Biagiola, Jose Luis Diaz, Rodrigo Goñi, Nahuel Otonelo, fermin rodriguez y Juan Manuel Calabia, giró en torno a la aplicación y propiedades de filtros lineales y no lineales, la correlación y convolución, y el uso de la transformada de Fourier para el análisis espectral de imágenes. El punto principal de la reunión fue la presentación de un trabajo práctico para implementar un algoritmo de detección de enfoque en videos utilizando los conceptos discutidos.

### Detalles

* **Filtros Convolucionales** lse posgrados introdujo los filtros convolucionales como un tema importante en el procesamiento de imágenes y redes neuronales. Explicó que los filtros lineales y no lineales se utilizarán para analizar la distribución de píxeles ([00:00:00](#_1hjdlig1c4cp)).
* **Análisis de Histogramas** Jonathan Borda compartió los resultados de una práctica en la que dos imágenes, una escala de grises y una flor, tenían histogramas idénticos a pesar de ser visualmente diferentes. lse posgrados aclaró que los histogramas muestran la distribución de niveles de intensidad, lo que puede ser engañoso si las imágenes difieren espacialmente pero comparten la misma distribución de intensidad ([00:02:16](#_9z3onkl0c9fv)).
* **Comparación de Histograma con Bins** Daniel Bazan agregó que al cambiar el número de "bins" de 32 a 64, los histogramas de ambas imágenes cambiaban, aunque los picos en el rango de 250 a 255 píxeles se mantenían consistentes. lse posgrados confirmó que con la misma cantidad de "bins", los histogramas siempre serían idénticos para este caso particular, ya que las distribuciones de intensidad eran iguales ([00:04:47](#_pcx2wp8n8aik)).
* **Distribución Espacial e Intensidad** Luciano Ceballos preguntó cómo los histogramas abordaban las "formas", y lse posgrados explicó la necesidad de una componente que considere la distribución espacial además de la intensidad ([00:04:47](#_pcx2wp8n8aik)). Se mencionó la transformada de Fourier como una herramienta para analizar el nivel de intensidad y la ubicación espacial, asegurando una representación única de las imágenes ([00:06:29](#_eqfvdn21mcqs)).
* **Cantidad de Bins en Histogramas** Erlin Rey cuestionó la cantidad de "bins" a usar en los histogramas, comparándolo con la granularidad en series de tiempo, y lse posgrados señaló que la elección depende de cómo se desean visualizar los datos ([00:07:45](#_g688g2hwjatm)). lse posgrados también anticipó el uso de histogramas de gradientes para incluir características espaciales ([00:08:57](#_uo8izl6zi3k4)).
* **Histogramas como Características (Features)** Sebastian Biagiola preguntó sobre la eficiencia de los histogramas como "feature" para identificar si una imagen es de día o de noche. lse posgrados respondió que, aunque pueden ser útiles en algunos casos (como clasificar imágenes claras/oscuras), los histogramas solos pueden llevar a entrenar modelos incorrectamente si la distribución de intensidades es similar entre imágenes distintas, a menos que el histograma caracterice la imagen de forma única ([00:10:17](#_y3vlc9h8mo5k)) ([00:13:23](#_ydo9iqsunfn6)).
* **Manipulación de Imágenes y Ruido** Jose Luis Diaz observó que la imagen degradada se había creado a partir de la otra para que coincidieran los histogramas. lse posgrados confirmó que ordenó los píxeles de la imagen original para crear el degradado. Explicó que, si bien en la práctica los histogramas no son idénticos, si son similares, pueden causar fallos en algoritmos de clustering o aprendizaje supervisado ([00:11:51](#_pcbdkj3imhni)).
* **Transformaciones de Imágenes con Operadores Lineales y No Lineales** lse posgrados revisó las operaciones de píxeles anteriores, como modificar brillo y contraste, y rotaciones o escalados ([00:13:23](#_ydo9iqsunfn6)). Destacó la necesidad de algo más allá de las operaciones lineales para tareas como la binarización adaptativa, que considera los píxeles vecinos ([00:14:44](#_bx23r0n8236y)). Mencionó aplicaciones como la restauración, corrección de defectos y extracción de características, y presentó ejemplos de cómo los filtros lineales pueden difuminar, enfocar o preservar los bordes de una imagen ([00:15:55](#_8i6bx9u17n0a)) ([00:18:33](#_wmen1c3c74ic)).
* **Filtros Lineales y la Operación de Correlación** lse posgrados detalló que los filtros lineales realizan una combinación lineal del píxel central con sus vecinos para obtener el valor de salida. Explicó que esta operación se realiza mediante una correlación en imágenes, usando un "kernel" o filtro, generalmente de punto flotante con ganancia unitaria (suma de coeficientes igual a uno) para evitar saturación de la luminancia ([00:19:58](#_x38lgsoc4ktq)) ([00:22:49](#_cah0d98p8pjz)).
* **Propiedades de los Filtros y Correlación** lse posgrados explicó que los filtros de desenfoque son filtros de paso bajo que atenúan las altas frecuencias (bordes) y dejan pasar las bajas frecuencias (transiciones suaves) ([00:21:21](#_ahcktd49nyu7)). Demostró la operación de correlación, donde el kernel se desplaza por la imagen, multiplicando y sumando los valores solapados para obtener el valor central de la salida ([00:22:49](#_cah0d98p8pjz)).
* **Stride, Padding y Origen del Kernel** lse posgrados definió "stride" como el desplazamiento del kernel (siempre uno para ellos) y "padding" como los ceros añadidos a los bordes de la imagen para que el kernel pueda operar en ellos y la imagen de salida mantenga el mismo tamaño ([00:24:02](#_jacm1jf4ltgm)). Jonathan Borda preguntó si los píxeles del borde desaparecerían sin padding, y lse posgrados confirmó que sí, lo que hace que la imagen se achique ([00:28:59](#_w5kqul8yfmxg)). lse posgrados también señaló que el origen del kernel está en su centro, a diferencia de una imagen ([00:27:45](#_4tovahiki2dd)).
* **Correlación vs. Convolución y la Delta de Kronecker** lse posgrados distinguió entre correlación y convolución, indicando que la convolución es más general y requiere espejar el kernel. Explicó que son equivalentes solo si el kernel o la imagen son simétricos ([00:30:13](#_f9dlmq9ob3ty)). Introdujo la delta de Kronecker como el elemento neutro de la convolución, comparable a multiplicar por uno, y que su convolución con una imagen produce la misma imagen o una desplazada ([00:31:48](#_cpisbwcmn1m7)).
* **Función Delta y Caracterización de Sistemas** lse posgrados mencionó que la función delta (de Dirac o Kronecker) contiene todas las frecuencias y es útil para caracterizar sistemas calculando su "respuesta al impulso" ([00:34:21](#_6zd1ihd8wgvu)). Rodrigo Goñi confirmó haber visto esto en ingeniería mecatrónica y mecánica vibratoria ([00:35:48](#_efc6kt9ias18)).
* **Filtros Predefinidos vs. Redes Convolucionales** lse posgrados contrastó el uso de filtros con coeficientes predefinidos para tareas como la extracción de bordes (que funcionan bien globalmente) con las redes neuronales convolucionales, que ajustan sus coeficientes para una tarea específica y generalizan mejor si el dataset comparte características ([00:35:48](#_efc6kt9ias18)). Explicó que para tareas simples, los filtros son más confiables y eficientes, mientras que las redes son costosas en recursos ([00:38:33](#_umy9ww92e9vv)).
* **Propiedades de la Convolución** lse posgrados enumeró propiedades de la convolución, como la conmutatividad y la posibilidad de sacar constantes multiplicadoras. Destacó la asociatividad, que permite combinar múltiples convoluciones en una sola para optimizar el procesamiento ([00:39:59](#_ndnuq6hgnroz)). También mencionó la invariancia en el tiempo, que significa que el orden de recortar y convolucionar no afecta el resultado, aunque puede haber variaciones en los bordes debido al padding ([00:41:24](#_49id17wewt57)).
* **Diferencias entre Correlación y Convolución** lse posgrados enfatizó que la transformada de Fourier de una correlación es diferente a la de una convolución, y que la correlación no es asociativa para operaciones con imágenes, a menos que el kernel o la imagen sean simétricos. Recomendó consultar la bibliografía para ver más ejemplos de filtros, como el kernel gaussiano y los filtros de Sobel y Prewit, que son importantes para extraer bordes ([00:42:34](#_gf934zk049zs)).
* **Funcionamiento de Filtros de Desenfoque y Detección de Bordes** lse posgrados explicó que los filtros de desenfoque (pasabajos) difuminan las transiciones abruptas (altas frecuencias), atenuando el ruido pero también los bordes de los objetos ([00:43:52](#_83hnz2j0d6aw)). Demostró cómo los filtros de detección de bordes, como los de Prewit, calculan la derivada de una imagen, resultando en valores altos en las transiciones (bordes) y cercanos a cero en regiones planas ([00:46:32](#_gydps6vfdssg)).
* **Gradiente y Características Invariantes** lse posgrados detalló que la convolución con filtros de borde es equivalente a calcular una derivada ([00:50:25](#_4n6om9o81i4l)). Explicó que el gradiente es ortogonal a la dirección del borde, permitiendo obtener el ángulo del gradiente a partir de las derivadas en X e Y ([00:51:36](#_38mh4hl9uwz8)). Resaltó que esta información es más abstracta y puede ser invariante a cambios en la intensidad o rotaciones de la imagen, siendo útil para algoritmos avanzados como Shift y ORB ([00:53:04](#_kmu0voeibtto)).
* **Filtro Laplaciano y Filtrado de Ruido** lse posgrados presentó el filtro Laplaciano como un extractor de bordes isotrópico. Advirtió que, al devolver el contenido de alta frecuencia, amplifica el ruido en la imagen ([00:54:39](#_ec2femvhdoax)). Por lo tanto, se recomienda aplicar un desenfoque gaussiano antes de la extracción de bordes o usar un kernel combinado, como el log
* **Extracción de Bordes con Kernels** Ise posgrados explicó las opciones para extraer bordes usando kernels como Laplaciano, Prewitt o Sobel. Destacó la posibilidad de trabajar con imágenes en formato "float" o convertirlas a "IGN 8" para obtener píxeles de borde, independientemente de la transición de luz a oscuridad ([00:58:48](#_hbws7050jd1q)).
* **Kernels Separables** Ise posgrados introdujo el concepto de kernels separables, que permiten transformar una convolución bidimensional en dos convoluciones unidimensionales para agilizar el proceso ([01:00:08](#_m22q93rb83ap)). Mencionó que kernels como el gaussiano son separables y que OpenCV los utiliza de esta manera, lo que optimiza el uso de memoria ([01:01:33](#_db2jkdo7rpi2)).
* **Unsharp Masking para Enfoque** Ise posgrados detalló el algoritmo de "Unsharp Masking", que genera un enfoque artificial en imágenes utilizando filtros gaussianos ([01:02:55](#_8eybgdqmx5q6)). El proceso implica filtrar la imagen, restar la versión filtrada de la original para crear una máscara de alta frecuencia y luego sumar esta máscara, multiplicada por una constante, a la imagen original ([01:04:30](#_5r1kqhcpuzct)). Este método potencia el contenido espectral de alta frecuencia para realzar detalles ([01:05:46](#_aipqaihj4hrl)).
* **Extracción de Bordes con Diferencia de Gaussianas** Ise posgrados explicó cómo usar kernels gaussianos para extraer bordes, siguiendo un principio similar al Unsharp Masking ([01:07:29](#_d9swh8mxljs5)). Al restar dos imágenes filtradas con gaussianas de diferentes sigmas, se obtiene una región de frecuencias medias que contiene los bordes y el ruido ([01:08:52](#_fp3vhyd7ru50)).
* **Filtros No Lineales** Ise posgrados abordó los filtros no lineales, señalando que no siempre tienen el concepto de coeficientes o convolución ([01:10:31](#_8ohkm97aonp)). Ejemplificó con los filtros de mínima, máxima y mediana, que operan seleccionando el valor mínimo, máximo o central de los píxeles solapados por un kernel ([01:23:50](#_gyb0eo3924m2)). Estos filtros son especialmente útiles para eliminar ruido como el "sal y pimienta", preservando los bordes y el contenido de la imagen de manera más efectiva que los filtros lineales ([01:28:05](#_5l8528hfzo5r)).
* **Implementación de Filtros en OpenCV y Supervisión** Ise posgrados mostró la implementación práctica de filtros en Python utilizando OpenCV y la biblioteca Supervision ([01:29:32](#_eweh9gaivgly)). Demostró cómo obtener kernels gaussianos y aplicarlos para desenfoque, así como el uso de filtros de mediana y la diferencia de gaussianas para la extracción de bordes ([01:31:09](#_wk9gjl1aoilp)) ([01:34:55](#_2z4q0lbvzl1c)).
* **Detector de Bordes Canny** Ise posgrados presentó el algoritmo de detección de bordes Canny como una forma más inteligente y confiable de extraer bordes ([01:36:19](#_18m4ojgu7pcg)). Explicó sus pasos clave: filtrado gaussiano para reducir ruido, cálculo del gradiente (módulo y ángulo) ([01:37:46](#_72a4gct9np1c)), supresión de no máximos para afinar los bordes ([01:39:06](#_6fx0jd8z6jlp)), y umbralización con histéresis para preservar bordes conectados y eliminar ruido ([01:40:29](#_eprewjri8my6)).
* **Introducción a la Transformada de Fourier** Ise posgrados comenzó la explicación de la transformada de Fourier, un concepto que permite obtener el contenido espectral de una imagen ([01:44:50](#_z0ab8se8jc9n)). Lo ilustró con señales unidimensionales para simplificar la comprensión, destacando cómo se representan la amplitud y la frecuencia de una señal en el dominio de Fourier ([01:46:25](#_lcmympexbxgt)). La transformada de Fourier representa una señal como una combinación de senos y cosenos, lo que permite visualizar su contenido espectral como "palitos" o envolventes de espectro para señales más complejas ([01:49:20](#_pnb7kdkv0otn)).
* **Propiedades y Aplicaciones de la Transformada de Fourier** Ise posgrados explicó que la transformada de Fourier de una imagen es única y es un buen descriptor para extraer características. Resaltó la propiedad de que la convolución en el dominio espacial es un producto punto a punto en el dominio de Fourier, lo cual es computacionalmente más eficiente gracias al algoritmo FFT ([01:52:13](#_rv9vqqakshm4)). En imágenes, las bajas frecuencias se encuentran cerca del centro del espectro, mientras que las altas frecuencias se encuentran más lejos, permitiendo la creación de filtros pasabajos y pasaltos mediante la manipulación del espectro ([01:55:02](#_2u0haqe22eud)).
* **Representación del Espectro en FFT** lse posgrados explicó que el espectro en FFT tiene una contraparte negativa y para una correcta visualización, es necesario aplicar un "FFT shift" para centrar el cero, especialmente en imágenes bidimensionales. Este paso es crucial para la interpretación gráfica de los resultados, aunque no es necesario para el filtrado ([01:58:51](#_v1z07bpqx592)).
* **Análisis de Señales e Imágenes en el Dominio Espectral** lse posgrados ilustró cómo una señal unidimensional y su contraparte en el dominio espectral pueden extenderse a imágenes, donde los picos representan máximos o mínimos de intensidad ([02:18:29](#_lp59d3u6fbmw)). Se mostró cómo se cargaron y graficaron imágenes de alta y baja frecuencia, calculando su transformada de Fourier y aplicando FFT shift para centrarla ([02:19:50](#_3jwrcodo4ymp)). Se observó que las componentes de alta frecuencia se alejan más del centro en la transformada, mientras que las de baja frecuencia permanecen cerca, y la presencia de armónicos en la transformada de Fourier ([02:21:20](#_xfuu1896bwea)). Nahuel Otonelo preguntó sobre la relación entre la señal senoidal y la imagen, y lse posgrados aclaró que las senoidales eran perfiles extraídos de las imágenes para ilustrar la intensidad de píxeles ([02:22:46](#_nelcnyz4s2jj)). Daniel Bazan también preguntó sobre la visualización de la imagen completa en contraste con las filas individuales ([02:26:36](#_vzu2jv99who7)).
* **Aplicación de Filtros en el Dominio Espectral** lse posgrados presentó la implementación de funciones para filtros pasa-altos y pasa-bajos utilizando formas circulares, que se aplican en el dominio espectral para modificar la imagen original ([02:27:53](#_uhwdbocdr0qv)). Explicó que el filtrado implica multiplicar las transformadas punto a punto y luego aplicar la transformada inversa de Fourier ([02:29:13](#_vcw3r77a9mje)). Se notó que los filtros diseñados sin considerar su existencia en el dominio espacial pueden generar artefactos como "ecos" en la imagen filtrada. Para evitar esto, lse posgrados explicó que se debe partir de un kernel que exista en el dominio espacial, como un kernel Gausiano, y expandirlo con ceros para que tenga las mismas dimensiones que la imagen original antes de aplicar la transformada de Fourier ([02:30:49](#_td10t33optx3)). Jose Luis Diaz señaló que el orden de complejidad del algoritmo de FFT es n log n, lo que lo hace muy rápido ([02:34:38](#_c0av7t5g3lle)).
* **Trabajo Práctico: Detección de Enfoque en Video** lse posgrados presentó el trabajo práctico que consiste en implementar un algoritmo de detección de enfoque en videos, similar a lo que hacen las cámaras. Se debe utilizar un paper provisto para calcular una métrica de enfoque para todo el frame y luego para una región más pequeña, opcionalmente calculando una matriz de enfoque ([02:36:04](#_zh9lunpm4d6e)). Se espera graficar la métrica en función de los frames para encontrar el punto de máximo enfoque y repetir el proceso con otra métrica de un segundo paper ([02:37:34](#_jt2ighiz2m4v)). También se plantea una opción para mejorar la región de enfoque con "sharpening" ([02:39:59](#_5ntzhlfs7l9k)). fermin rodriguez y Juan Manuel Calabia preguntaron sobre la entrega y la posibilidad de trabajar en grupo, a lo que lse posgrados confirmó que todos los TPs se entregan por medio de un link a un repositorio y se pueden hacer en grupo ([02:41:08](#_fj8avkt06up6)). Daniel Bazan también preguntó sobre el trabajo en grupo. Juan Manuel Calabia preguntó sobre un cuestionario final, a lo que lse posgrados confirmó que se realizará en la última clase ([02:42:15](#_eip743pwampv)).

### Pasos siguientes recomendados

* fermin rodriguez will upload the project to a repository and send the link via email.

*Revisa las notas de Gemini para asegurarte de que sean correctas.* [*Obtén consejos y descubre cómo toma notas Gemini*](https://support.google.com/meet/answer/14754931)

*Danos tu opinión sobre el uso de Gemini para tomar notas en una* [*breve encuesta.*](https://google.qualtrics.com/jfe/form/SV_9vK3UZEaIQKKE7A?confid=0HlMYqqdkHioQX739GrhDxIVOAIIigIgABgBCA&detailid=unspecified)

📖 Transcripción

11 sept 2025

## Visión por Computadora I (3/8) - a213b25 - Transcripción

### 00:00:00

**lse posgrados:** Hola, ¿cómo están? ¿Todo bien?  
**fermin rodriguez:** Buenas,  
**Jose Aviani:** Buena, ¿cómo va,  
**lse posgrados:** Bueno, eh la idea de la clase de hoy es ver por ahí uno de los temas más importantes, ¿sí? Que hay tanto en lo que es procesamiento de imágenes como redes neuronales para procesamiento de de imágenes en general. A ver, ahí está. Eh, ¿qué es? Eh, filtros convolucionales. Sí, son filtros lineales. La idea es también ver un poquito algo de filtros no lineales. Eh, yo les había pedido que para hoy tengan hecha la segunda parte, ¿no?, de lo que fue el primer el primer trabajo práctico. Eh, bueno, selecciono alguien al azar, ¿sí? y me cuenta que qué fue lo que hicieron. Eh, eh eh bueno, está Jonathan Borda.  
**Jonathan Borda:** ¡Uf! Qué suerte. ¿Cómo va? ¿Se escucha?  
**lse posgrados:** Sí, sí, se escucha, se escucha.  
**Jonathan Borda:** Eh, un segundo que que estoy buscando el repositorio.  
**lse posgrados:** Sí, no hace falta, eh, digamos, no hace falta.  
**Jonathan Borda:** No sé si mostrar.  
 

### 00:02:16

**lse posgrados:** Sí, o sea, puedes comentar qué fue lo que qué fue lo que hiciste, qué fue lo que viste y qué conclusión puedes sacar al respecto. No.  
**Jonathan Borda:** Sí, teníamos dos imágenes. En una imagen se veía como una escala una escala de colores de blanco a negro, una escala gris y la otra era una flor. Y cuando hicimos el histograma se veía que los dos gráficos eran iguales. pusimos 50, por ejemplo, y se veían iguales. Entonces, en principio, digamos, después preguntaba si eso se podía usar como feature o no. El principio es como que es medio engañoso porque uno puede decir parece ser la misma imagen, pero los píxeles están distribuidos diferentes. Entonces ahí no aportaba mucho. Eh, y después eh lo intentamos hacer de otra forma, ahora no me acuerdo bien cómo se llama el método, pero que ayudaba un poco más a a ver lo que sería la parte espacial, ¿no? según se llamaba así, que te va a otro histograma y y sí empezaba a ser diferente. Entonces ahí sí aportaba un poco más de información.  
**lse posgrados:** Okay, claro. Muy bien. El histograma en sí te habla de la distribución, pero en términos de niveles de intensidad, o sea, vos podés tener exactamente la misma cantidad de píxeles que valen 100, 50, 20, 30 en una imagen y en otra.  
 

### 00:03:47

**lse posgrados:** Pueden ser dos imágenes completamente diferentes, pero los histogramas pueden ser iguales, digamos.  
**Jonathan Borda:** Mhm.  
**lse posgrados:** Si la si la cantidad de píxeles de determinado valor es la misma, la distribución en términos de niveles de intensidad eh puede ser la misma. Sí, es es raro que sea exactamente la misma. En este caso, para estas dos imágenes sí lo es, digamos, son dos histogramas idénticos, punto a punto, porque la distribución en términos de nivel de intensidad eh, bueno, es es igual en ambos en ambos casos. Okay, okay. Bueno, bien. ¿Alguien más hizo algo distinto?  
**Jonathan Borda:** Yeah.  
**lse posgrados:** No sé, quiere compartirlo que era un ejercicio bastante pavo igual, ¿no? Pero digamos la la idea, la conclusión acá que que llevó al compañero es correcta. Alguien por ahí lo hizo y dijo, "No, mira, yo lo hice y a mí no me dan iguales o algo por el estilo." O todos más o menos llegaron a la misma a la misma conclusión.  
**fermin rodriguez:** Sí, sí.  
**Daniel Bazan:** Sí, lo que nosotros habíamos visto era la comparación de B también, eh, cuando cambiábamos de 32 a 64, si no recuerdo mal, ahí se veía como eh iba cambiando el histograma.  
 

### 00:04:47

**lse posgrados:** Bien.  
**Jose Aviani:** Oh.  
**Daniel Bazan:** Eh, entiendo que por cómo agrupaba los píxeles, la cantidad de píxeles que se agrupaba y en amb sí se marcaba en ambos el pico que hacía en desde 250 a 255 de de píxeles de esas imágenes. En el resto se con 64 se veía más uniforme la distribución de píxel.  
**lse posgrados:** Sí, eso digamos analizando cada histograma por por separado, pero en realidad eh digamos si vos hacés un histograma con x cantidad de beans y el otro y para la otra imagen usas la misma cantidad de binans, eh los citamas van a ser idénticos, digamos, no puede dar distinto nunca. para esto, para este caso en particular. Sí, Luciano, no termino de entender la pregunta, digamos, eh, ¿a qué te referís con las formas?  
**Luciano Ceballos:** No, claro. Eh, porque en los historiamas estos hablan de la intensidad de los píxeles, pero no habla de las formas, digamos. O sea, son idénticos en cuanto a la distribución de los de las intensidades, pero claro, habría que analizarlo desde otra perspectiva, digamos, desde cómo se qué formas van formando estos píxeles o no se debería como complementarlo.  
 

### 00:06:29

**lse posgrados:** Claro. O sea, faltaría como una componente una componente que te hable de la distribución espacial, no solo de la distribución en términos de niveles de de intensidad.  
**Luciano Ceballos:** Ah, claro. Sí, sí. Bien. Eso, supongo que lo veremos en esta clase clase siguiente, eso, digamos.  
**lse posgrados:** Sí. Lo que lo lo que lo que relaciona el nivel de intensidad y la ubicación espacial eh es la transformada de Julier. Sí, porque podés analizar algo eh desde otro punto de vista.  
**Luciano Ceballos:** Mm.  
**lse posgrados:** Y bueno, algo que que tiene la la transformada Furier es que eh las transformaciones son únicas, ¿no? Para que dos imágenes tengan o dos funciones tengan la misma transformada de Fulier, tiene que ser la misma la misma imagen, digamos. Y nosotros podemos pensar una transformada a una transformada de furier de una imagen como bueno, podemos trabajar con el módulo de la transformada de furier de una imagen y bueno, pensarlo o analizarlo de la misma manera que analizaríamos  
**Luciano Ceballos:** Sí.  
**lse posgrados:** un histograma 2D, por ejemplo, que nosotros pas vieron que nosotros algo hemos visto, ¿sí? Bien, hacíamos un histograma 2D y teníamos una imagen.  
 

### 00:07:45

**lse posgrados:** Bueno, eso podría ir a un clasificador o algún algoritmo de clasering o lo que sea y eh nos aseguramos de que la representación es única, ¿sí? No, no voy a tener dos dos transformadas iguales para imágenes distintas.  
**Luciano Ceballos:** Bien, bien. Bueno, gracias.  
**lse posgrados:** Eh, alguien más había levantado la mano. Puede ser. Eh, Erlin apareces acá en en la fila.  
**Erlin Rey:** Sí, sí, sí. Igual creo que algo ya estuviste mencionando porque era un poco ese análisis de los histogramas y los distintos biníamos tomar en términos de qué era lo que tenía sentido o que no, porque una de las cosas era analizar eh cuánto era la cantidad, ¿no? Entonces tirábamos distintos valores como para ver cómo se modificaba los histogramas de ambos, más allá de que dieran iguales por la imagen que es sintética para que dieran más o menos lo mismo. Es bueno, desde la perspectiva ya más allá de un análisis a futuro, ¿qué es lo que tiene sentido? El 256 para tener más datos, una compresión un poco más con 128 64. Creo que cuando tiraste lo de la transformada de Furier me parece que va más por ahí que simplemente analizar el histograma, pero no sé si entendí mal.  
 

### 00:08:57

**Erlin Rey:** Amén.  
**lse posgrados:** No, a ver, cuando analizas un histograma, la cantidad de BINS la la pones vos, o sea, es como cuando analizas cualquier tipo de de distribución, eh, es como, no sé, tenías que analizar lo mismo cuando analizas, no sé, datos tab de un CCB. ¿Qué qué granularidad te sirve? Te sirve saber poner, no sé, tenés algo de, no sé, una serie de tiempo. Te sirve saber, no sé, qué sucedió en cada milisegundo de tu serie de tiempo o por ahí querés saber qué pasó en cada minuto o por ahí querés saber qué pasó en cada hora. Entonces, con las imágenes es lo mismo. O sea, vos te sirve saber cuántos píxeles hay que valen 0 1 2 3 4 c o por ahí capaz lo querés agrupar en oscuro, gris y claro. Sí. Entonces ahí depende depende de un poco de cómo visualizas los datos. Eh, después vamos a ver que hay hay una forma de usar histogramas como feature por ahí, no necesariamente histogramas eh de niveles de intensidad, sí, sino que son histogramas de gradientes, porque bueno, justamente uno puede calcular el gradiente en una imagen y ahí tiene características espaciales también, de hecho lo vamos a arrancar a ver la clase de hoy.  
 

### 00:10:17

**lse posgrados:** Eh, sí, Sebastián.  
**Sebastian Biagiola:** Sí, justo te iba a comentar algo, algo similar y eh en la consulta, en la pregunta también eh decía de posible fecture para un dataset, tener el histograma. Eh busqué y leí por ahí que tal vez puede servirnos de forma rápida para identificar eh si una imagen, por ejemplo, está es de noche o es de día. ¿Eso sería eficiente? ¿No sería eficiente?  
**lse posgrados:** Eh, o sea, leías clasificar si si la imagen tipo es oscura o es clara, decís vos.  
**Sebastian Biagiola:** Claro, buscarle una Claro, exactamente.  
**lse posgrados:** Asumo que es de noche con apagadas.  
**Sebastian Biagiola:** Buscarle una por qué lo usaríamos en un dataset a estos histogramas si perdemos esta información espacial que que es útil.  
**lse posgrados:** Sí, la la pregunta igual era si los histogramas son útiles como feature, digamos. Eh, esa era la creo que la pregunta la pregunta exacta. En algunos casos sí, en otros casos no, para este caso no y para todos los casos donde tengo imágenes distintas, pero con porque no hace falta que sea igual, ¿no?, la distribución de niveles de intensidad. con que sea parecida, eh, por más que las dos imágenes sean distintas y pertenezcan a objetos distintos, si la distribución de los niveles de intensidad es similar, eso ya alcanza para entrenar un modelo y que me dé todo mal, porque digamos, no hace falta que sean iguales, eh, punto a punto.  
 

### 00:11:51

**Sebastian Biagiola:** Entiendo, entiendo.  
**lse posgrados:** Bueno, eh, ¿alguien más tiene alguna consulta? Eh, Erlin, vos con la mano levantada, pero no sé si quedó levantada o tenés otra otra consulta.  
**Erlin Rey:** No, no, quedó levantada. Quedó levantada y la saco. Perdón.  
**lse posgrados:** Ah, bueno, no, no está bien.  
**Jose Luis Diaz:** Muy rápido. Claramente esto creaste imagen, la imagen de Shad a partir de la otra imagen para que coincida, no sería mucha casualidad otra forma.  
**lse posgrados:** Sí, no, claro, yo lo que hice fue ordenar los los píxeles de creo que de menor a mayor o de mayor a menor, o sea, agarré la y por eso por eso quedó quedó con el degradé porque están  
**Jose Luis Diaz:** Yeah. Ah, buena onda, hermoso.  
**lse posgrados:** ordenados, digamos, de esa forma. Pero sí, la la imagen es la misma. Igual si si uno agarra esa imagen, la ordena, le desarma los píxeles y se le suma una constante o algo por el estilo, eh el histograma puede dar parecido, ¿sí? O sea, no exactamente igual, pero puede dar parecido y se da eso de que comenté recién de que capaz en la práctica no vas a tener dos histogramas idénticos, pero sí similares y cuando, no sé, hagas un algoritmo de clasering o algo de aprendizaje supervisado solo basado en en histogramas, tanto el clustering como algo supervisado va a fallar, salvo que el histograma tenga algo que caracterice de manera única la imagen que hay casos, digamos, se puede  
 

### 00:13:23

**lse posgrados:** llegar a dar, tipo, no sé, si tienes que hacer un clasificador de flores, por ejemplo, donde todas las flores tienen colores distintos, eh en esos casos podría ser un buen un buen feature. Pero bueno, depende depende mucho de de la aplicación. Y como les digo, hay hay el histograma se usa, pero eh no no el histograma solo de colores, sino que un poquito modificado. Bueno, nosotros estuvimos viendo las dos clases anteriores, operadores de de píxeles, ¿sí? Podíamos hacer cosas como esta, ¿sí? O sea, o sea, teníamos una imagen, le podíamos modificar el brillo en este caso, eh podíamos modificar el contraste, podíamos podíamos hacer varias varias cosas. Eh, también podíamos trabajar no sobre el valor del píxel, sino sobre dónde está ubicado y empezar a sea a hacer cosas como estas o aplicar rotaciones o escalados. Sí, por ahí no le dimos mucha bola en la práctica, pero estaban las matrices de transformación para este tipo de de operaciones. Eh, pero ya para el final de la clase vimos que, claro, si yo quiero hacer otras cosas, ¿sí? Eh, necesito necesito algo más, o sea, no me alcanza solamente con operaciones lineales.  
 

### 00:14:44

**lse posgrados:** Vimos que, bueno, para binarizar, por ejemplo, hay casos donde seleccionar un umbral único no es suficiente. O sea, el umbral único, ¿qué hace? Mira píxel a píxel y se fija si es mayor o menor que que el umbral que yo puse. Y uno termina usando umbrales adaptativos que los calcula no solo a partir del valor del píxel que quiero que quiero ver si si lo mando al cer o al 255, sino que a partir de los vecinos de ese de ese píxel, ¿sí? Y bueno, esos vecinos como que la región donde está ubicado el Pixel empieza a tener eh cierto protagonismo también, lo cual tiene sentido porque en general si yo agarro cualquier píxel de acá y seguramente eh comparta características con sus vecinos cercanos, ¿sí? O sea, los ya sean los ocho que tiene pegados o capaz eh los que están los pegados a los que tiene a los que a los primeros ocho. Sí. Entonces, bueno, uno puede expandir esa región tanto como uno quiere, obviamente no mucho, porque si yo parto de acá y la empiezo a expandir, va a llegar un momento que no sé, termino agarrando píxeles de acá y píxeles de acá que no tienen nada que ver.  
 

### 00:15:55

**lse posgrados:** Pero si analizo un píxel y sus ocho vecinos inmediatos o los inmediatos a esos ocho, probablemente compartan características similares. Eh, y bueno, hay aplicaciones, sí, acá hay una lista, ¿no?, de solamente algunas, eh, que, bueno, requieren otro tipo de operaciones para poder para poder llevarlas a cabo. Sí, restauración, corrección de defectos, bueno, extracción de de características. Yo por ahí no quiero trabajar con el nivel de intensidad del píxel. sino que quiero algo más. Por ahí me gustaría laaburar con eh la orientación del gradiente sobre ese píxel. Por ejemplo, acá vemos que hay un degradé eh y tenemos un gradiente que avanza, por ejemplo, en esta en esta dirección, ¿sí? Porque esa es la dirección de crecimiento de la intensidad de los niveles de de los valores de los píxeles. Sí. y para el otro lado decrece. Eh, lo mismo podríamos mencionar, por ejemplo, por acá, donde también tenemos una transición de luz a a oscuridad, como así medio en diagonal. Y ya voy adelantando temas, ¿no? Un poco más adelante. Si tengo una región que es monótona, ¿sí? Por ejemplo, esta acá no tengo no tengo un gradiente, acá el gradiente vale vale cero, ¿sí?  
 

### 00:17:15

**lse posgrados:** Entonces me van a interesar este tipo de regiones donde por ahí hay transiciones de luz oscuridad. ya sean así medio como medio globales o grandes, como también cosas locales, ¿no? Porque acá mismo, ¿no? La sombra de de o el tono de piel acá de Arnold Schwarzenegger tiene como pequeños gradientes. Si yo empiezo a mirar ventanas chiquititas, ¿sí? Todo esto son características que, bueno, me sirven para caracterizar a una a una imagen. Eh, y hay muchas muchas aplicaciones más. Sí. Por ejemplo, esto que está relacionado con el TP de la clase de hoy, ¿sí? Eh, vamos a ver eh operaciones que nos permiten desenfocar una imagen, ¿sí?, de forma artificial, pero también enfocarla, ¿sí? Dado una imagen que está un poco desenfocada, yo puedo aplicar una serie de de algoritmos para enfocarla un poquito. Sí. Y esto digamos se corre en eh los teléfonos celulares o las cámaras que son muy muy económicas porque no les da no le da para hacer foco en todo, porque no hay una parte móvil que se pueda que se pueda desplazar como en una cámara convencional, sí, que uno puede agarrar una lente tele y moverla de allá para acá.  
 

### 00:18:33

**lse posgrados:** Entonces, tanto el desenfoque como el enfoque eh puede ser en parte físico, pero eh se ayuda con algoritmos. Sí, justamente para para emular efectos tanto de enfoque como de desenfoque. Y bueno, la primer familia de como de elementos que nos permite hacer este tipo de cosas son los filtros lineales. Acá esta es la imagen original y acá tenemos tres versiones de la imagen procesada con filtros lineales. ¿Sí? Y vemos que, bueno, podemos difuminar una imagen, eh, podemos enfocarla, ¿sí? O podemos hacer algo intermedio. Fíjense que acá la imagen está difuminada y los bordes de la imagen también difuminados. Sí, acá tenemos todo lo contrario. Los bordes están mucho más acentuados, ¿sí? Y acá tenemos como algo intermedio, o sea, tenemos una imagen que a priori preserva los bordes de la imagen original, pero tiene la textura de los objetos está difuminada. Sí, es como una especie de blur, pero sin tocar los los bordes de una de una imagen. Eh, y a diferencia por ahí de de las operaciones que nosotros vimos la clase pasada, para poder llevar a cabo esto, yo lo que hago es hacer una una combinación, ¿sí?  
 

### 00:19:58

**lse posgrados:** De hecho, es una combinación lineal de eh el píxel que quiero modificar con sus vecinos, ¿sí? A partir de del resultado de hacer esa combinación, yo termino obteniendo el valor que va a tener el píxel que quiero modificar en la imagen de de salida. ¿Sí? O sea, que acá el valor de salida de un píxel va a depender en parte de sí mismo, pero en parte también de los píxeles que están en la misma región de la cual yo saqué al al píxel que quiero modificar. Eh, bueno, ejemplos, ¿no? Nosotros vimos eh algoritmos de binarización que hacían algo por el estilo. Eh, bueno, los ajustes adaptativos del histograma no lo vimos en la teórica, creo que lo mencioné nada más, pero lo vimos en la práctica, donde calculábamos una función de transformación para cada región adentro de la imagen. Bueno, código LBP lo vamos a ver dentro de un par de clases, ¿sí? Y eh, ¿cómo hacemos esto? Bueno, mediante en imágenes, ¿sí? Mediante una operación de correlación, ¿sí? Vamos a superponer un elemento que vamos a llamar kernel o filtro. Sí, acá lo hice de 3x 3, pero puede tener cualquier cualquier tamaño, ¿sí?  
 

### 00:21:21

**lse posgrados:** Eh, y vamos a hacer una correlación entre todos los elementos del filtro con todos los elementos de la imagen que están solapados con el filtro. Sí, ahora en la próxima diapo va a quedar un poco más claro, pero digamos la expresión matemática es esta. Tengo que hacer una suma de productos. Sí, multiplico punto a punto este valor con el que tiene abajo, este valor con el que tiene abajo, este con el que tiene abajo y lo sumo todo. Sí. Eh, ¿por qué se llaman filtros? Bueno, esto viene de, en realidad del procesamiento de de señales unidimensionales, ¿sí? donde uno aplica una serie de operaciones para filtrar determinadas frecuencias, filtrar o o incluso amplificar. Sí. Eh, siempre que tenemos filtros de desenfoque, lo que estamos haciendo es aplicar un filtro pasabajos. ¿Sí? Todo lo que son transiciones suaves en una imagen se lo considera baja frecuencia. Todo lo que son transiciones abruptas o bordes o cambios de contraste, eh son frecuencias altas. ¿Sí? Entonces, un filtro que desenfoca una imagen, lo que va a hacer es difuminar los bordes, ¿sí? Va a atenuar la alta frecuencia y va a dejar pasar la baja frecuencia.  
 

### 00:22:49

**lse posgrados:** Por ahí alguien que que estudió electrónica o o computación, algo ya habrá habrá visto. Igual vamos a ver un par de un par de ejemplos más, ¿sí? O sea, lo que hacemos cuando aplicamos una correlación con con filtros en imágenes es alterar el contenido espectral de una de una imagen. Sí, el contenido espectral son las frecuencias que la que la compone. Bueno, acá por ahí queda un poco más claro, ¿no? La operación que tenemos que hacer. Tenemos acá nuestro kernel que lo vamos a ir paseando, o sea, lo vamos a desplazar por toda la imagen. Y para setear el valor central, lo que tengo que hacer es multiplicar todos, sumar, ¿sí? Y lo pongo ahí. Eh, ahora el kernel o el filtro no es una imagen. Sí, porque una imagen para nosotros tiene píxeles que tienen un tipo de dato de unsign int, pero el kernel está en punto flotante. Sí. Acá tenemos números con coma. Eh, para nosotros van a tener todos los filtros una característica, ¿sí? Y es que van a tener ganancia unitaria, ¿sí? O sea, la suma de todos los coeficientes tiene que ser igual a uno.  
 

### 00:24:02

**lse posgrados:** Porque si es mayor a uno, lo que estoy haciendo es eh amplificando. Y es medio peligroso eso porque me puedo llegar a ir eh por encima de 255, ¿sí? lo cual no quiero. Entonces, justamente para evitar que se sature la luminancia de la imagen, vamos a trabajar siempre con kernels unitarios, o sea, la suma de todos vale vale uno. Eh, bueno, acá un par de definiciones que por ahí si si cursaron la materia de de deep learning, por ahí la ya las conocen, ¿sí? Eh, el striite es la cantidad de píxeles que yo voy a desplazar mi kernel al momento de pasearlo por la imagen, tanto en X como en Y. Esto cuando uno diseña redes convolucionales es un hiperparámetro, eh, un hiperparámetro de cada capa. Para nosotros el stripe siempre va a ser uno, ¿sí? Aunque nada nos impide eh digamos eh codear nuestro propio algoritmo de correlación donde es distinto de uno. Sí, en Open CB por defecto eh siempre es el stride en uno para las convoluciones, pero digamos la operación es bastante sencilla, es una sumatoria dentro de un loop. Sí, digamos, si lo quisiéramos modificar podemos podamos poner nuestro propio strike.  
 

### 00:25:15

**lse posgrados:** Y después tenemos el concepto de padding. Sí puede, ¿qué pasa? Esta operación de sumas de productos yo la puedo hacer solamente si el kernel está solapado con la imagen, pero no sé si yo alíneo el píxel central del kernel con, no sé, con con este píxel de acá, bueno, con este, bueno, esta es la imagen de salida. ¿Qué pasa acá? El kernel está solapado, pero esto se queda afuera. Esto se queda fuera. Sí. Entonces tengo 1 2 3 cu cinco píxeles donde no puedo hacer esta esta operación, ¿sí? Porque se está solapado con con la nada misma. Solamente tengo estos cuatro. Entonces, eh a priori yo solamente podría hacer esta operación de de filtrado cuando tengo un coverage completo entre el kernel y la imagen, con lo cual la imagen de salida termina siendo más chica que la imagen de de entrada. Sí, si yo no tengo si yo no pongo algo acá para poder apoyar el kernel y hacer la la cuenta, eh, eso que yo pongo es justamente el padding, ¿sí? Son números que yo me invento para poner a los costados de la imagen y que el kernel esté solapado, incluso estando en los bordes por ahí.  
 

### 00:26:37

**lse posgrados:** Acá en un kernel de 3x 3 tengo que agregar una fila y una y una columna en cada en cada borde. O sea, tendría que agregar algo acá, algo acá, algo acá y algo acá. Pero si el kernel fuera de 5\* 5, por ejemplo, y acá ya tendría que agregar dos en cada lado. Si fuera de 7\* 7 tendría que agregar tres y así. Sí, obviamente cuanto más grande es el kernel, más padding tengo que agregar para preservar el tamaño de de la imagen. Sí, idealmente yo convoluciono una imagen que tiene una resolución y me gustaría tener a la salida eh bueno, una imagen de la misma resolución, ¿no? Que se vaya achicando. Sí. En donde sí se da eso de que la imagen se va a chicando. Es en redes neuronales donde nosotros hacemos convulsiones, aunque algunas sí son con compadding, ¿sí? Eh, la mayoría no. justamente porque yo lo que quiero es pasar de una imagen a algo sintético que por ahí tiene un décimo de la resolución de la imagen, pero mantiene ciertas características. ¿Sí? Eh, para nosotros el padding sí siempre va a ser cero a los costados.  
 

### 00:27:45

**lse posgrados:** De esa manera es como la forma más eh neutral de no modificar eh mucho la imagen de salida. O sea, simplemente se van a anular los coeficientes que no están solapados y hago la cuenta de la sumatoria y los productos para los píxeles que sí se solapan, pero como agregué los ceros, puedo apoyar el kernel en un borde. Sí. Eh, lo mismo, esto en redes neuronales convolucionales también es un hiperparámetro de cada capa. Eh, bueno, y después el origen del kernel. Ya habíamos dicho que no es una imagen. Si en una imagen el origen estaría acá. El origen del kernel es acá en el centro. ¿Sí? Con lo cual, el píxel central es el píxel de origen, es el píxel que yo voy a modificar o el píxel que yo tengo que alinear en realidad con el píxel que quiero modificar en la imagen. Si, de hecho, fíjense acá que está resaltado el centro del kernel con este píxel color verde que es el 01, es el 22, ¿sí? en este caso y vemos que esta operación va a setear el valor del píxel 22 en la imagen de de salida.  
 

### 00:28:59

**lse posgrados:** ¿Sí? ¿Alguna pregunta de de esto? Bien. Sí, me pareció que alguien abrió ahí el micrófono.  
**Jonathan Borda:** Sí, sí. Eh, entonces si no hay paddil, los del borde nunca van a cambiar porque están muy al borde.  
**lse posgrados:** Los del borde nunca van a cambiar. Los del borde van a desaparecer directamente porque ponerle si no hay padding vos solamente vas a poder hacer la cuenta para Claro, en el caso de un de un filtro de 3x 3 a partir de la fila dos.  
**Jonathan Borda:** Claro, a partir de la fila dos o columna dos.  
**lse posgrados:** Sí. y la columna dos también. Con lo cual vos decís, "Bueno, voy a setear este valor en la imagen de salida y este valor, bueno, este valor de acá, que es el primero con el cual vos podrías hacer la cuenta, va a definir este píxel en tu imagen." O sea, los de los costados eh desaparecen. Eso si no hay si no hay pading, o sea, no es que no cambian, sino que desaparecen directamente. Y si empezas a convolucionar una imagen sin padding una y otra vez, te terminas quedando sin imagen a larga, que de hecho así es como laburan las las redes convolucionales, empiezan a a chicar.  
 

### 00:30:13

**Jonathan Borda:** Yeah.  
**lse posgrados:** Bueno, nosotros hablamos de eh correlación. De hecho, eh eso de multiplicar y sumar es justamente una una correlación, es la definición de de una correlación. Eh, pero si lo queremos ver desde un punto de vista más general, tenemos que pasar a la convolución que es parecida. Sí. solamente que hay un pequeño cambio. En imágenes a veces se habla tanto de correlación como convolución, como si fuera lo mismo. Y en filtros convolucionales, en muchos filtros convolucionales es lo mismo, pero eh digamos matemáticamente son dos operaciones distintas, muy distintas. Eh y si el filtro no cumple cierta propiedad y yo aplico correlación, el resultado va a dar mal. Sí, porque la convolución y la correlación solamente son iguales cuando una de los dos uno de los dos elementos, el kernel o la imagen, son simétricos. La imagen difícilmente sea sea simétrica. Sí, tendría que ser como un patrón geométrico, pero el carnel sí. De hecho, fíjense que acá hay ejemplos de filtros eh bidimensionales y todos tienen una característica, ¿no?, de que este elemento es igual a este, este elemento es igual a este, este elemento es igual a este, este es igual a este, ¿sí?  
 

### 00:31:48

**lse posgrados:** Con lo cual hay una simetría al igual que acá y al igual que acá, ¿sí? Digamos, al ser simétricos, la correlación y la convolución son lo mismo. Sí, en la práctica, ¿no? Eh, porque yo podría cambiar el kernel. De hecho, vamos a ver que hay kernels que no son simétricos y ahí se rompería todo. Entonces, la operación que realmente va es la convolución, por más que en el 99% de los casos nos dé el mismo resultado. Hay un par de propiedades bastante copadas, ¿sí? Eh, ah, ¿en qué cambia la convolución de la correlación? Eh, tengo que espejar uno de los dos elementos. Es más fácil espejar el kernel antes que espejar la imagen. ¿Sí? Entonces, lo que hago es este este espejado, ¿sí? Si este es mi kernel, cuando convoluciono, bueno, pasaría a ser este mi kérnel, ¿sí? Y después es lo mismo, suma sumatoria de de productos y desplazamiento. La convolución tiene un elemento neutro, ¿sí?, que es la delta. Eh, en el mundo continuo es la delta de Draak, ¿sí?, que es una función que tiene ancho cero, pero área uno, ¿sí?  
 

### 00:33:00

**lse posgrados:** O sea, es una función de distribución. Y en el mundo discreto, ¿sí? es la delta de Chroneker, que es un palito en en uno. Sí, básicamente. O sea, tenemos acá nuestra recta y en, perdón, en cero tenemos un palito que vale que vale uno. Si yo convoluciono una imagen con una delta, obtengo la imagen. ¿Sí? Si yo convoluciono la imagen con una delta desplazada, voy a obtener un desplazamiento de la imagen. O sea, es como que esta es mi imagen, la convoluciono con una delta que está, no sé, en uno, por ejemplo, y lo que voy a obtener es digamos que esta es mi imagen. Sí, voy a obtener la imagen pero desplazada, ¿sí? Corrida de lugar. Eh, si convoluciono una delta con una imagen o la imagen con una delta, obtengo siempre el mismo resultado. Sí, justamente porque es el elemento neutro comparable a una multiplicación por por uno. Eh, no sé si hay mucho más para comentar. Unidad, sí, hay un montón de cosas, pero la idea por es no irnos tanto por las ramas con teoría de señales.  
 

### 00:34:21

**lse posgrados:** ¿Vieron alguna vez una una función delta, así digamos en probabilidad de estadística o alguna materia de de grado? Análisis 3, análisis dos. Seguramente algo de esto se ve. Es una es una función primero bastante particular, o sea, algo que tiene área cero, pero integra uno. Es algo que debería debería llamar un toque de la atención. Y después tiene ciertas propiedades, ¿si? Eh, por ejemplo, eh, si uno calcula el contenido espectral de una delta, resulta ser que contiene todas las frecuencias, ¿si? O sea, todas las frecuencias posibles están contenidas en algo que tiene ancho cero y área y área uno. Entonces, como tiene todas las frecuencias, es una es una función bastante útil para caracterizar sistemas. ¿Sí? Si yo quisiera saber eh cuál es la función transferencia de un sistema, algo que podría hacer es calcular su respuesta a la delta, ¿sí? O sea, convoluciono eh y obtengo la respuesta. También se conoce como la respuesta al impulso. Eh, ¿lo vieron esto alguna vez? Matemática avanzada. Vieron convolución. ¿Es una materia de acá del posgrado o bien?  
 

### 00:35:48

**Rodrigo Goñi:** No, no es una materia de ingeniería mecatrónica universidad de Claro.  
**lse posgrados:** Claro, debe ser tipo análisis análisis tres o algo o algo por el estilo.  
**Rodrigo Goñi:** Sí. En mecánica vibratoria también vimos eh función delta para sistemas discretos y también sí.  
**lse posgrados:** Claro. Bueno, en en supongo que vieron eso, vieron lo que sería la función de heavis, también la función escalón para calcular la respuesta al escalón, digamos. Son dos funciones típicas para para hacer identificación de de sistemas. Bien. Bueno, eh acá hay ejemplos, ¿sí?, de algunos filtros. La idea ahora es ver un par de de ejemplos más, ¿sí? Algo por ahí que caracteriza el procesamiento con filtros convolucionales y que lo diferencia de hacer procesamiento con redes convolucionales es que con filtros básicamente yo digo, bueno, paraá, quiero, por ejemplo, extraer bordes de una imagen, o sea, dado una imagen quiero ver su versión, pero solamente con bordes. Sí. Bueno, hay filtros que se encargan de extraer bordes, hay filtros que extraen todos los bordes, hay filtros que extraen bordes en X, bordes en I, eh, y que funcionan relativamente bien para todas las imágenes del mundo, ¿sí? O sea, agarro cualquier imagen, la convoluciono con un filtro que extrae bordes y voy a extraer, voy a obtener los bordes.  
 

### 00:37:15

**lse posgrados:** Sí, por ahí voy a querer mejorar un poco la solución, voy a poder jugar con el tamaño del filtro. Capaz que voy a tener que preprocesar la imagen previamente porque tiene ruido. Entonces, por ahí la voy a le voy a aplicar un filtro que extrae ruido y después extraigo los bordes. Sí, pero nosotros vamos a trabajar con filtros que ya tienen los coeficientes predefinidos. Sí, en redes neuronales convolucionales lo que uno hace es parte de la imagen que quiere procesar, parte de un ejemplo de esa imagen ya procesada y le pide a la red que ajuste los coeficientes de su filtro, de sus filtros, eh, para que pueda realizar esa opción. ¿Sí? Entonces, si yo después le muestro una imagen que de alguna manera comparte características con el dataset que se usó para el entrenamiento, eh, bueno, voy a obtener un resultado relativamente similar. ¿Va a servir eso para todas las imágenes del mundo? Y no. Sí. O sea, lamentablemente no, porque cualquier imagen que está por fuera de la distribución del conjunto de entrenamiento va a dar mal, pero sé que puede llegar a dar bien para imágenes que comparten características con el conjunto de entrenamiento, ¿sí?  
 

### 00:38:33

**lse posgrados:** O sea, una red convolucional lo que hace es ajustar los coeficientes para llevar a cabo una tarea. Por eso, obviamente, hay tareas muy simples como extracción de bordes o extracción de otro tipo también de características que eh es mucho más fácil y mucho más eh confiable hacerlo con filtros, ¿sí?, que hacerlo con una red eh convolucional. Obviamente hoy en día hay modelos mucho más avanzados que una red convolucional. Sí. Eh, o por ahí que que parte de su de su background son redes convolucionales, pero en realidad son un poco más que pueden llegar a generalizar y hacer tareas muy copadas, por ejemplo, modelos de difusión estable o o por ahí cosas basadas en Transformers, tipo modelos generativos. Pero bueno, el tema es que ahí eh sias un cluster con 5 GPUs para poder correrlo, bueno, depende del modelo igual, ¿sí? Eh, y para operaciones simples por ahí podés llegar a obtener algo comparable a un filtro, pero bueno, a un costo mucho más alto. Sí. Entonces, siempre hay que tener en cuenta qué es lo que uno quiere lograr y ver por ahí por dónde lo ataca, si usa redes neuronales, si usa filtros, por ahí una combinación de de las dos cosas. Por ahí extrae características con filtros y o con algún algoritmo basado en filtros y después esas características sí van a una red neuronal.  
 

### 00:39:59

**lse posgrados:** Sí, depende mucho ya de del problema y de los recursos que uno tiene, porque también no es lo mismo tener a su disposición, no sé, eh GCP o o AWS con créditos ilimitados o tener un, no sé, un Cortex M4 donde tengo que cuantizar la red para que anden 4 bits. Sí, digamos, es como que depende también del mundo en el cual uno se esté se esté moviendo para resolver el problema. Eh, algunas propiedades que tiene la la convolución, ¿sí? Al ser lineal, eh bueno, es conmutativo, es lo mismo convolucionar el kernel con la imagen que la imagen con el kernel. Esto ya lo comenté. Si hay una constante multiplicando el kernel o multiplicando la imagen, la puedo sacar para fuera. Y por ahí esta es como la más importante. Si yo tengo toda una sucesión de operaciones lineales, eh creo que podrí y en un momento, no sé, por ahí tengo dos filtros que se están convolucionando y que después convolucionan a a la imagen o tengo la imagen convolucionada con un filtro y después otra vez convolucionada con el otro. Lo que puedo hacer es sacar esto, convolucionarlo afuera del pipeline, convertirlo en un kernel número tres, eh, y usar esto para no tener que hacer estas dos convoluciones que no tienen mucho sentido porque van a dar el mismo número, ¿no?, al ser operaciones lineales.  
 

### 00:41:24

**lse posgrados:** Entonces, podría convolucionar afuera, obtener los coeficientes y ahí sí hago pasé de dos convoluciones en rindime a una sola, lo cual va a ser más rápido. Ahí está mucho, la verdad que no no nos interesa. Sí, o sea, que sea invariante en el tiempo. Lo que me dice es que convolucionar la imagen entera con un filtro y después recortar un pedazo de esa imagen es lo mismo que recortar un pedazo de la imagen y convolucionarlo. Sí. Eh, el resultado va a ser el mismo, obviamente va a variar en los bordes si uno eh convoluciona así como vimos acá. Sí, justamente porque, bueno, en los bordes se agrega padding y puede ser que el valor cambie. Pues bueno, depende del tipo de padding que yo agregue. Si no agrego padding, me va a quedar una imagen con dos o tres tiras de píxeles menos. Eh, vamos a ver que también podemos convolucionar en el dominio espectral. Ahí es como otro paradigma, ¿no? No existe el concepto de de padding como lo vimos acá. Eh, pero bueno, nada, simplemente eso, o sea, recortar un pedazo de la imagen y convolucionarla es lo mismo que convolucionar toda la imagen y después recortarla.  
 

### 00:42:34

**lse posgrados:** Y bueno, acá la principal diferencia entre correlación y convolución, sí, primero es que la transformada de furier de una correlación es diferente a la de una convolución, o sea, se mapean a dos operaciones diferentes, eh, y que la correlación no es asociativa para operaciones con imágenes. Si no se cumple esto que vimos acá, obviamente, ¿en qué caso se va a cumplir? Y cuando el kernel o la imagen sean simétricos, entonces ahí sí es es la misma operación en el fondo. Bueno, acá hay más ejemplos de filtros por ahí. Para esto sí les recomiendo ver alguno de los libros de la bibliografía por ahí, no para profundizar en lo que es la convolución, sino para ver qué filtros hay, porque acá hay un par, ¿sí? Pero hay muchos más, ¿sí? O sea, si yo quiero extraer bordes o quiero difuminar algo, si bien estos son por ahí los los más usados, hay hay muchos otros más que son como pequeñas variantes, ¿no?, de estos de estos kernels. Por ahí algo que vamos a usar bastante es el kernel gauciano, ¿sí? que sale de eh hacer un muestreo de una gausiana en dos en dos dimensiones y los filtros de Sbel.  
 

### 00:43:52

**lse posgrados:** Acá faltó el SEL para poder este está en X y en Igual que que los prwirven para extraer bordes en I y bordes en en X. Este kernel de acá, sí, junto con el kernel gauciano son filtros de desenfoque o filtros pasabajos. Veamos por qué son filtros pasabajos. Sí. Somos que yo tengo una imagen que tiene este perfil que podría ser eh tranquilamente que esto esta región de acá es toda gris y no tiene puntitos blancos. Entonces tengo algo que tiene un tono y de pronto sube, ¿sí? Porque era gris. Un gris puede ser 100, 150 y después pasa a valer 255 y se queda en 255. Digamos podría ser tranquilamente este perfil si yo lo trazo acá. Al convolucionarlo, por ejemplo, con un kernel que tiene todo unos acá es todo uno, divide un noveno, ¿no? Para para que sea eh unitario o convolucionado con un kernel gauciano. Lo que me va a dar es algo así, ¿sí? O sea, el perfil resultante lo que va a hacer es difuminar un poco esta transición entre lo gris y lo y lo blanco. Sí, esto tiene todo lo que lo que sea abrupto, tanto la delta como un escalón, tiene muchas frecuencias.  
 

### 00:45:17

**lse posgrados:** Sí, tiene un contenido espectral bastante bastante ancho si uno lo quiere lo quiere graficar y en la práctica lo vamos a a graficar. Todo lo que es un degradé tiene un contenido de espectral acotado. ¿Sí? Y al difuminar algo, ¿sí? Lo que hacemos es atenuar el contenido espectral en las altas frecuencias preservando solamente las bajas. ¿Sí? Lo mismo sucede acá. Por ejemplo, este perfil podría ser lo que tenemos acá en esta zona, ¿sí? Acá en el centro del cuadrado hay un puntito blanco, ¿sí? Entonces yo trazo el perfil y lo que voy a ver es todo gris, todo gris, todo gris, puntito blanco, todo gris, todo gris, todo gris. Sí. Eh, al convolucionar una región que tiene algo así con un kernel de desenfoque, lo que hago es bajar el nivel de de intensidad de del puntito de la zona clara a costa de eh bueno, expandir un poco la región. Sí, esto lo podemos pensar como como un ruido que está en la imagen. Entonces, yo al convolucionar lo que hago es bajar el nivel de intensidad del ruido. Sí, porque fíjense que esto acá se mantiene constante, ¿eh?  
 

### 00:46:32

**lse posgrados:** Pero eh bueno, un precio a pagar porque estos filtros no saben diferenciar lo que es ruido de lo que es el borde de un objeto. Y un borde que sí, el borde es una transición de oscuridad a luz o luz oscuridad, entonces es una región de alto contraste. Entonces, así como esto atenúa el ruido, también atenúa las regiones de alto contraste, eh, haciendo que, bueno, la imagen termine siendo una imagen de bajo contraste, difuminando las características que esa imagen posee. ¿Sí? Bueno, esto es para lo que eran los filtros de desenfoque. ¿Cómo funcionan los filtros de detección de bordes, ¿sí?, que son estos, este de acá, este de acá y todos los que tengan esta esta pinta, ¿sí?, de valores negativos y valores positivos en el medio. Eh, bueno, vamos a hacerlo acá. Agarremos este que es fácil, ¿no? Tiene todo unos y todo menos unos. Si estoy en esta región, ¿sí? Y quiero hacer la convolución, ¿sí? O sea, en esta región, digo, esta este pedazo de la diapositiva donde está todo blanco, tengo -1 -1 -1 0 1 1 1. Quiero convolucionar acá.  
 

### 00:47:42

**lse posgrados:** Vamos a hacerlo a mano. Tengo que hacer cuánto valdría un píxel en esta en esta región. O sea, tiene tiene una imagen que está que es toda blanca, vale 255. Entonces vemos que todos los píxeles que están solapados con mi kernel valen 255 porque están todos en la región blanca.  
**Jose Luis Diaz:** 255.  
**lse posgrados:** Entonces tendría que hacer -1 \* 255 + -1 \* 255 + -1, ¿sí? \* 255 + 1 \* 255 + 1 \* 255 + 1 \* 255, o sea, que tengo -25 - 255 - 255 + 255 + 255 + 255 me termina dando cero. Sí, lo cual tiene sentido porque es una región que no tiene bordes, tiene una región perfectamente plana, sin textura. Entonces acá va a dar cero. ¿Y dónde va a dar cero o valores cercanos a cero? Bueno, cuando porque no no todas las imágenes tienen valores constantes, digamos. Pero si tengo una región que es más o menos eh monótona, ¿sí? donde no hay grandes cambios eh de iluminación de color, de textura, capaz no da cero, pero me va a dar un número muy chiquito, ¿sí?  
 

### 00:49:08

**lse posgrados:** Porque tengo cosas negativas de un lado, cosas positivas del otro y se va a terminar cancelando o digamos cuando haga la suma me va a dar un número bastante pequeño. ¿Sí? ¿En dónde esto va a dar un número relativamente grande? Y cuando esté en una región de transición, ¿sí? supamos que estoy acá y tengo acá tengo mi región donde todo vale 255 y acá tengo una región donde todo vale cero, ¿sí? O sea, estoy en una región oscura y bueno, yo estoy desplazando mi kernel, ¿sí? Y llego acá y tengo 1 que va a estar en la región de los 255. Acá tengo los ceros, estos no me importan. Y acá tengo -1, -1, -1. ¿Y qué voy a obtener? Y voy a obtener un 255 \* 3 por estos unos que están adentro de la región, ¿sí? Más -1 \* 0 -1 \* 0 -1 \* 0 me da 0. ¿Sí? Entonces, acá en la región de transición, en todos los píxeles que están en el borde, yo voy a obtener, no sé, como 700 y pico, ¿sí?  
 

### 00:50:25

**lse posgrados:** O sea, voy a obtener un valor altísimo. Eh, y ese valor va a cambiar de signo si yo me estoy moviendo de oscuridad a luz o de luz oscuridad, ¿sí? O sea, que no solo obtengo el borde, digamos, la ubicación del borde, porque este este número tan grande va a ir a parar al píxel que corresponde en una imagen de borde, eh, sino que también puedo inferir en cómo fue la transición, si me moví de una zona oscura a una clara o de una zona clara a una zona oscura. Por ahí acá, acá lo podemos ver, ¿no? Tenemos una imagen que tiene las dos transiciones. Pasamos de luz oscuridad y después de oscuridad a luz tenemos un perfil que es este perfil de acá. Y cuando calculamos su versión de de bordes, ¿qué es lo que obtenemos? Y en la región donde no hay cambios, ¿sí?, de intensidad, sea todas estas regiones, esto vale cero. Pero cuando hay un cambio de luz oscuridad, tenemos máximos, ¿sí? O mínimos, tenemos extremos. Y hacer una convolución con un filtro de borde, en especial los filtros de Prewit, eh no es otra cosa que calcular una derivada.  
 

### 00:51:36

**lse posgrados:** Sí. O sea, uno cuando trabaja en tiempo discreto tiene varias formas de calcular una derivada y la forma en la cual se aproxima la derivada de imágenes, ¿sí? es tomando los valores extremos, ¿no? O sea, vos tenés una sucesión de de valores, entonces quiero calcular la derivada en este punto y lo que terminas haciendo es restar este con este. Sí, hay otra forma de de aproximar derivadas que es restar este con este y otra que es restar este con este. Es como que uno aproxima hacia adelante o aproxima hacia atrás o aproxima hacia los dos hacia los dos lados. Sí. Solo que en imágenes yo no solo no derivo en una dimensión, derivo en dos. Entonces tengo tanto una derivada en X como una derivada en i. Por eso tenemos filtros que extraen bordes en X y extraen bordes en I. Y algo interesante es que el gradiente sí siempre va a ser ortogonal a la dirección del borde. Entonces, si yo obtengo las derivadas en x de una imagen, que no no es otra cosa que convolucionar la imagen con un filtro prewit en X y las derivadas en i, o sea, o tengo dos imágenes de borde para a partir de una imagen, eh, podría obtener el ángulo de o la dirección del gradiente, ¿sí?, a partir del ángulo en cada uno de los píxeles de la imagen, o sea, con el arcotangente, ¿no?, de de I  
 

### 00:53:04

**lse posgrados:** sobre X. Y esto está bastante bueno porque uno parte de una imagen extrae características como son los bordes, que los puede usar ya para tareas de de clasificación o detección y puede ir aún más allá que es extraer la imagen de ángulos, ¿sí?, a partir de de las dos imágenes de borde. Y ahí yo tengo información que obviamente está fuertemente ligada a cómo están distribuidos los niveles de intensidad de una imagen, pero lo tengo de una forma un poquito más abstracta, ¿sí? Porque por ahí en términos de intensidades y si yo te cambio la iluminación puede ser que eh las características desde el punto de vista de niveles de intensidad y cambien para dos objetos que capaz son iguales. Eh, desde el punto de vista espacial, los X sí capaz que se mantienen, pero lo que seguro se va se va a mantener, ¿sí? Eh, son los ángulos, ¿sí? Incluso si yo roto la imagen, yo podría tomar ángulos relativos a una posición. Entonces ahí eh tengo una característica que es invariante a cambios en los niveles de de intensidad, pues para calcular un ángulo tengo que hacer un arcotangente de un cociente. Así que si las proporciones son las mismas, el ángulo va a dar el mismo. Y si yo tomo un ángulo relativo respecto de algún otro ángulo dentro de la imagen y lo mantengo, soy invariante a rotaciones.  
 

### 00:54:39

**lse posgrados:** De hecho, es el principio de funcionamiento de algoritmos como Shift, ORB y similares que vamos a ver dentro de dentro de un par de clases. Tenemos más filtros. Sí. Eh, así como tenemos un kernel gauciano, tenemos un Lllaciano. El Laplaciano también es un filtro que extrae bordes, pero es isotrópico. O sea, no lo puedo separar en bordes en X y bordes en I. es como que extrae bordes en todas las en todas las direcciones. Bueno, acá tenemos un pequeño desarrollo, ¿no? O sea, este es el la la expresión de un lapllaciano. Sí, lo puedo las derivadas segunda la aproximo de esta de esta forma, lo escribo en forma matricial y me armo este este kernel. Eh, tiene un problema, no solo el aplaciano, sino todos los filtros que extraen bordes. Y es que al extraer bordes uno lo que hace es eh devolver el contenido de alta frecuencia de una imagen. ¿Sí? El tema es que el ruido en imágenes está mayoritariamente del lado de la alta frecuencia en el espectro. ¿Sí? O sea, ¿qué es el ruido? Bueno, por ejemplo, esto de acá podría considerarse ruido, o sea, son puntos de alta intensidad en algo que por ahí tiene una intensidad mucho más baja.  
 

### 00:56:04

**lse posgrados:** Entonces, eh, claro, todo esto en una imagen, si yo extraigo bordes de acá, por ejemplo, lo voy a ver recontraamplificado. Eh, entonces lo que uno suele hacer siempre es aplico desenfoque gauciano y después extraigo bordes. Como es una operación tan común y bueno, estos son operadores, operaciones lineales, lo que se hace es directamente trabajar con un nuevo kernel que es el logciano del gauciano. ¿Por qué? ¿Qué haríamos? Bueno, agarro la imagen, la convoluciono con un kernel gauciano para atenuar el ruido y después aplico en la placiano. Bueno, esto es una operación lineal, esto es una operación lineal. Lo que hago es combinar el placiano con el kernel gauciano haciendo una convolución. forma un nuevo kernel que es el log. Sí, acá lo tenemos dibujado. Y eh esto que ya es un kernel con nombre, apellido y sus coeficientes lo convoluciono con con la imagen. Sí, todos los kernels que nosotros vimos o prácticamente todos eh si bien los podemos crear y podemos hacer una convolución con alguna biblioteca en OpenCB también están implementados como funciones. O sea, hay una función que es eh get ganan kernel, se la pido, me da el kernel gauciano.  
 

### 00:57:19

**lse posgrados:** Hay funciones para aplicar la convolución directamente. Sí, ganan blur, por ejemplo, que uno le pasa la imagen y obtiene imagen convolucionada sin tener que tocar el kernel siquiera. Sí, pero hay casos donde yo necesito el kernel para hacer operaciones por ahí un poco más complejas o para o por ahí para combinar el kernel con otra cosa. Ahora vamos a ver cómo cómo trabajamos eso porque no es tan simple. Bueno, acá algo que ya les mencioné, ¿no? Tenemos una imagen con ruido. Eh, trazamos un perfil y si bien vemos estamos acá el escalón que pasa de oscuridad a luz, tiene como un pastito, ¿no? Porque, bueno, tanto del lado donde todo o mayoritariamente todo vale 255, tenemos unos puntitos y también tenemos puntitos del otro lado. Cuando yo derivo o cuando yo aplico un filtro de bordes, este sería el perfil resultante. ¿Sí? Y si bien yo puedo ver una transición, fíjense que acá esto es como mucho más denso que en otros lados, eh, es medio medio indistinguible. Sí, lo sé yo, porque medio lo puedo lo puedo mapear a a la imagen de arriba. Sí, por eso siempre es importante aplicar un filtrado de ruido antes de hacer una una extracción de de bordes.  
 

### 00:58:48

**lse posgrados:** Eh, bueno, aquí hay ejemplos porque también cuando uno extrae bordes con el kernel de la plaiano, con Prewid o con Svel, tenemos como dos opciones. La primera es trabajar con la imagen en el tipo de dato resultante que es float, porque el kernal está en float. Yo hago la operación de convolución y obtengo una imagen que puede tener valores positivos o valores negativos o valores en punto flotante, porque, bueno, eso me va a dar lo que ya les comenté, información de cómo fue la transición de oscuridad a luz o luz oscuridad para cada para cada píxel de borde. Esto capaz que no me es tan útil, ¿sí? Eh, entonces lo que hago es pasarlo a IGN 8, pero por ahí el objetivo es extraer píxeles que se correspondan con bordes, más allá de si eh ese borde está en la frontera de algo oscuro a la derecha y algo claro a la izquierda. Sí, por ahí eso no me importa. Yo solo quiero extraer bordes porque después voy a hacer algún análisis morfológico, voy a seguir haciendo algún tipo de procesamiento para entonces pues entonces parametrizar esos bordes, por ejemplo. Sí, esto de la parametrización de bordes la vamos a ver eh la clase que viene. Sí, pues fíjense que acá nosotros digamos entramos con una imagen y salimos con otra imagen.  
 

### 01:00:08

**lse posgrados:** como que está bien, es una imagen que ya es un descriptor porque tiene de alguna manera la información un poco más sintetizada que la imagen original, pero eh yo no sé si el borde es una línea recta, si es una curva, cuánto mide. Sí, obviamente lo puedo saber si miro la imagen y no sé, la miro yo con contando los píxeles, pero no tengo no tengo todavía eh una info una info clara, sí, pero sí tengo algo que es el punto de partida para obtener eh esto de lo que de lo que estamos hablando. E bueno, esto por ahí después lo vamos a ver un poquito más en detalle en la práctica también o por lo menos el ejemplo, ¿no?, de cómo generar estas estas imágenes. Bueno, esto lo menciono muy rápidamente. Sí, existe un concepto, ¿sí? Que es el de los kernels separables. ¿Por qué? Y uno puede separar un kernel, ¿no? No siempre. Si el kernel cumple con ciertas propiedades, ¿cuáles son las propiedades? Son las que están acá. Sí. eh para transformar una convolución bidimensional entre el kernel y la imagen en dos convoluciones unidimensionales. ¿Sí?  
 

### 01:01:33

**lse posgrados:** Esto está bueno porque permite agilizar un poco este proceso de de sumatorias y multiplicaciones. Eh, el kernel separable básicamente se separa en dos vectores. Entonces, cuando yo multiplico esos dos vectores, obtengo la matriz. Para obtener esos dos vectores, tenemos este apartado de acá. A mí mucho no me interesa, sí, que que lo sepan, pero lo que sí me interesa es que sepan que los kernels, por ejemplo, el kernel gauciano es un kernel separable y Open CB hace uso de esto para hacer las convoluciones. Y si uno le pide a OpenC un kernel gauciano, no te va a dar una matriz. Sí. Por ejemplo, yo le pido a Openbche, dame un kernel gauciano de eh dimensión si en lugar de obtener una matriz de 7\* 7, uno obtiene un vector, ¿sí? en R7, porque Open CB asume que el que usa el framework conoce el procesamiento de imágenes y sabe que, claro, el kernel gauciano es un kernel separable, entonces lo puedo caracterizar por un vector y un vector de siete posiciones ocupa menos espacio de memoria que una matriz de 7\* 7. Y si yo quiero obtener la matriz, lo que haría es, bueno, multiplicar el vector por el su traspuesto y después normalizar.  
 

### 01:02:55

**lse posgrados:** para que tenga ganancia unitaria y ahí obtengo un kernel gauciano. Sí, nada, por eso está esta diapositiva acá. Después, eh, esto es algo que ya lo fuimos comentando, ¿no? O sea, en función de cómo son las transiciones en una imagen, al aplicar un filtro de bordes, yo puedo estar en estas situaciones, ¿no? Puedo tener gradiente cero en alguna de las de las direcciones. Siempre el gradiente va a ser perpendicular a la región del borde. Si son transiciones estrictamente horizontales o estrictamente verticales, se va a anular una de las direcciones. En general voy a estar en estos casos, ¿sí?, donde voy a tener un gradiente no nulo en ambas, ¿sí? Este, porque bueno, puedo tener degradezés o transiciones en cualquier eh en cualquier con cualquier orientación. ¿Alguna consulta de esto? nos queda una dos diapositivas más de filtros lineales. Ahí podemos y ahí hacemos un intervalo cortito y y ya vamos a una a una práctica de filtros lineales. Bueno, si no hay si no hay dudas seguimos. Un short masking es un algoritmo para hacer enfoque, ¿sí? O sea, genera un enfoque artificial en una imagen, pero lo hace a partir de filtros gaucianos, que son filtros de desenfoque.  
 

### 01:04:30

**lse posgrados:** En realidad, ¿sí? ¿Cómo funciona esto? Bueno, veamos primero cómo funciona. Después explico por qué funciona. Primero agarramos la imagen y la filtramos con un kernel gauciano. Eh, después agarramos esa esa imagen sin filtrar y le restamos su contraparte filtrada. Sí, de esa manera obtenemos como una especie de máscara y después lo que hacemos es sumar esa máscara, ese residuo multiplicado por una constante a la imagen original. ¿Sí? ¿Qué qué es lo que tiene esa máscara? Si nosotros hacemos un diagrama de BODE, ¿sí? Los diagramas de BODE lo que hacen es, bueno, el módulo, el el BODE para para lo que sería el módulo, lo que me da es el contenido espectral. Sí. Eh, la imagen sin procesar podemos asumir que es una cosa así, ¿sí? O sea, tiene un contenido espectral bastante bastante amplio al aplicar un filtro pasabajos, a ver si esto me deja cambiar el el color. Al aplicar un filtro pasabajos, lo que hacemos es atenuar toda esta zona, ¿sí? Quedándonos con este contenido espectral.  
 

### 01:05:46

**lse posgrados:** ¿Sí? Bueno, al restar el contenido espectral original con esta con esta parte, lo que estaríamos obteniendo es esto. Si claro, o sea, yo tengo la imagen filtrada, tengo la imagen original, al restar obtengo la parte verde, que es la componente de alta frecuencia en esa imagen. Y recuerden que la alta frecuencia es lo que me da eh bueno, la percepción de de bordes, la percepción de de enfoque. Obviamente esto funciona para ciertos casos, ¿no? Si yo tengo una imagen toda blurriada, por más que lo haga, no no me va a salir. Y al sumar, va, perdí, perdí acá el puntero. Al sumar, sí, ese este residuo que me queda, pero multiplicado por una constante, lo que estoy haciendo de alguna manera es potenciando el contenido espectral de la alta frecuencia. ¿Sí? Si hacemos el gráfico de la suma, me quedaría esto y esto así, donde volvemos de vuelta a la parte roja. Esto es lo que la imagen ya tenía. Sí, esto no se toca y lo verde es lo que se le estaría agregando multiplicado por una constante. Vamos a ver que eh esto funciona, ¿sí?  
 

### 01:07:29

**lse posgrados:** Eh, en casos donde yo todavía estoy a punto de difuminarme, pero estoy en la frontera, de hecho, bueno, por ahí en el TP va a quedar va a quedar mucho más claro, ¿eh? Pero bueno, acá tenemos la imagen original, ¿sí? Acá tengo K = 1, ¿sí? A medida que aumento el K, podemos empezar a ver cada vez más detalles, ¿sí? Y son detalles que en realidad siempre estuvieron en la imagen. Yo no estoy creando nada nuevo, simplemente estoy amplificando ciertas ciertas regiones, ¿sí? Pero lo hago desde el contenido espectral de la imagen y no desde los niveles de intensidad. Sí, obviamente esta esta constante no puede ser muy grande porque esto puede saturar. Sí, de hecho, si empezamos a analizar el histograma, vamos a ver que esto se empieza a saturar en algún momento. Sí, ahora en la práctica le podemos poner valores de K gigante si vemos que qué es lo que me da. Eh, bueno, así como podemos usar kernels gaucianos para enfocar una imagen, también podemos usar kernel gaucianos para generar eh extractores de bordes. El mismo concepto, ¿sí? El kernel gauciano se caracteriza por obviamente su tamaño y después por el sigma que es el parámetro, ¿no?, que tiene que tienen todas las aucianas.  
 

### 01:08:52

**lse posgrados:** Eh, y hay algo que por ahí me olvidé de mencionar, a mayor sigma más se difumina una imagen. Entonces, si yo tengo dos filtros con sigmas distintos, uno más grande que el otro, eh lo que estaría haciendo es, bueno, difuminar más o difuminar menos. Y siguiendo el mismo principio del del algoritmo de masking, si yo filtro una imagen con una gaciana de un sigma alto y después la misma imagen, sí, la filtro con una gaciana de un sigma abajo y las resto entre sí, lo que estaría haciendo es restar algo que tiene esta frecuencia de corte con algo que tiene esta frecuencia de corte, lo que va a ser que eh sobreviva esta región. Sí. Y esa región, si bien no es un exactamente un pas altos, termina siendo como un filtro pasabanda y el ruido va no solo el ruido, sino que los bordes están dentro de esta región de frecuencias medias. De hecho, esta es la imagen eh filtrada con este método. Sí. Fíjense que aparecieron un montón de puntos de ruido que por ahí finalmente no se veían. Y esto es eh con un procesamiento que vamos a ver ahora en la práctica, ¿sí? O sea, a pesar de obtener un resultado así que parece ruidoso, en realidad lo podemos lo podemos mejorar y tener una una idea de cuáles son los bordes de esta imagen.  
 

### 01:10:31

**lse posgrados:** Bueno, y nos quedan filtros no lineales para después de del intervalo. No sé si antes alguien tiene alguna alguna consulta para hacer. Yo sé que por ahí estos temas son medio complicados. Sí, yo la verdad que no no pretendo que entiendan la parte eh de contenido espectral porque ni siquiera forma parte de de la materia. Es algo de eh por ahí temas mucho más avanzados de de matemática, análisis 3, procesamiento de señales y sistemas. Eh, y si bien sirven para entender las cosas de una manera por ahí un poco más profunda, eh, a fines prácticos, creo que la teórica y lo que vamos a ver en la práctica cubre todo lo que uno tiene que saber para más o menos entender en qué situación usar cada cosa. Bueno, si no hay consultas, yo diría que volvamos 8:25. Hacemos un un intervalo medio medio cortito. Sí. Así que 8:25 estamos de regreso. Bueno, seguimos seguimos un toque más. Eh, la idea es ver un poquito lo que son los filtros eh no lineales y después hacemos una una práctica. Eh, a diferencia por ahí de los filtros lineales, ¿sí? En los filtros no lineales, por un lado, no siempre existe el concepto de de coeficientes.  
 

### 01:23:50

**lse posgrados:** ¿Sí? Y eh no siempre existe el concepto de convolución. De hecho, tenemos acá dos creo que todos estos ejemplos, sí, eh, son filtros que no tienen coeficientes. Sí, en lo que serían los filtros de mínima y máxima, sí existe el kernel como un objeto para decir, bueno, tengo algo que tiene una dimensión y voy a analizar lo que sucede con los píxeles que están eh superpuestos con ese con ese kernel. Sí. Pero no hay no hay coeficientes porque en el caso del filtro de mínima lo que hago es agarrar todos los píxeles que están solapados y devolver el mínimo. Sí, en el caso del kernel de máxima, lo mismo, pero devuelvo el valor máximo. Lo podemos relacionar a una especie de min y max pulling, por ejemplo. Sí, si ya cursaron aprendizaje profundo. Entonces, claro, acá no tiene sentido hablar de de kernels, pero sí de de tamaño, porque no es lo mismo analizar el mínimo de un kernel que tiene nu píxeles que un kernel que por ahí, no sé, tiene 7 por 7 o 11 por 11 o cualquier tamaño. Sí, el caso de la mediana es parecido, tampoco existen los coeficientes acá, ¿sí?  
 

### 01:25:13

**lse posgrados:** Y en vez de buscar el máximo o el mínimo, lo que hago es es calcular la mediana. O sea, agarro todos los píxeles, los ordeno y te doy el que está el que está justamente en el en el medio. Eh, hay algo interesante acá, ¿sí? En especial, por ejemplo, voy a tomar el de la mediana como ejemplo y es que, por ejemplo, para una mediana yo no tengo que conocer eh o mejor dicho, el orden original de los de los valores no influye en el cálculo de la de la mediana, ¿sí? O sea, vos puedes tener algo que sea 1 2 3, algo que sea 1 3 2 o algo que sea dos eh 3 o que que me queda 3 2 1 y no sé si no me estoy salteando alguna otra combinación, eh, pero la mediana de esto, esto, esto y esto va a valer siempre lo mismo. Sí, se ve eso. O sea, la distribución espacial de los niveles de intensidad no no me afecta al cálculo de la de la mediana. Eh, o sea, que solamente le importa el valor, ¿no?, dónde está ubicado. Tenemos un una familia de de de kernel, bueno, en todos los kernels convolucionales en realidad, en especial en el kernel gauciano donde se pondera la posición.  
 

### 01:26:42

**lse posgrados:** Sí, un kernel gauciano. Acá tenemos un ejemplo, está sin normalizar. De hecho, podemos ir para atrás, así agarramos un kérel gauciano de verdad. Siempre vamos a ver esto, ¿sí? Donde en el centro es donde tengo la mayor ponderación y a medida que me alejo del centro, fíjense que el valor de los coeficientes decrece, con lo cual un kérnel gauciano eh pondera por ubicación. Cuanto más eh cerca del centro estoy, cuanto más cerca estoy del píxel que estoy analizando, más peso le doy. Entonces existe algo que se conoce como el filtro de mediana ponderada, en donde yo multiplico punto a punto, ¿sí? Esto no es una convolución, sino que es una multiplicación de eh una ventana con coeficientes gaucianos por la porción de la imagen que quiero modificar. Y a esa yo le calculo la la mediana. Parece una pavada, pero ahora vamos a ver eh cuál es el efecto de hacer de hacer esto. Sí, porque lo que hace es darle un poco más de valor al píxel central por por la multiplicación y penaliza, ¿sí?, a los píxeles que están lejos de del centro. Estos filtros no lineales se usan para eh eliminar ruido de una imagen.  
 

### 01:28:05

**lse posgrados:** Tenemos esta imagen que se le agregó un tipo de ruido que se conoce como saltan pepper, que son puntos blancos y puntos negros. Eh, bueno, por ahí lo primero que podría hacer es filtrar con desenfoque uniforme, que es el kernel ese que tenía todo unos dividido en un noveno o con un kernel de desenfoque gauciano. Sería un efecto similar. Eh, y bueno, está bien, se se atenúa, sí, pero se difumina un poco y no solo se difumina el ruido, sino que se difuminan los bordes de la imagen. Yo esto lo podría seguir filtrando para bajar aún más estos estos valores, pero se me estaría amorfando un montón de características que tiene la imagen. Con los filtros de mínima y máxima obtengo este efecto, ¿sí? O sea, con unos eh elimino puntos negros, con los otros puntos blancos. más o menos preservo los los bordes, pero se destruye un poco eh el contenido de la imagen. Ahora, con un filtro de mediana, solo de mediana, logro este efecto. Sí. Y con uno de mediana ponderada, logro este que parece casi casi perfecto. Si uno se pone a analizar en detalle va a ver que hay cosas un poquito raras, pero a priori eh el filtro de mediana y el filtro de la mediana ponderada son la mejor opción para eh este tipo de de ruidos.  
 

### 01:29:32

**lse posgrados:** Sí, se termina devolviéndote una reconstrucción casi casi perfecta. Y esto funciona porque, claro, el ruido así de este tipo nunca va a estar en el nunca va a estar en el medio, o sea, siempre van a ser valores extremos. Entonces, claro, cuando cae superpuesto con el kernel y voy a terminar devolviendo el valor del medio antes que un 0 o un 255. Sí entendió más o menos esta esta idea. Bueno, podremos pasar a a la práctica a nivel eh implementación o desde el punto de vista práctico.  
**Jonathan Borda:** Sí.  
**lse posgrados:** La verdad que la creo que el tema de filtros es muy muy simple, pero sí hay como mucho contenido conceptual para entender. Bueno, acá no voy a usar Mat Blot Lip, si para esto voy a usar Supervision, que es una biblioteca que tiene un montón de cosas, entre otras cosas eh tiene un graficador bastante piola que tiene matlot lip por debajo y está pensado para trabajar con imágenes que ya están en BGR, o sea, asume que lo que vos le pases está en en BGR, por lo menos a priori. Y bueno, sirve simplemente para ahorrarnos un par de líneas de código al momento de hacer un plot de una de una imagen. Acá lo que hice fue extraer un kernel gauciano, pedirle a OpenC un kernel gauciano.  
 

### 01:31:09

**lse posgrados:** Le digo, "Bueno, dame uno de tamaño 5 de sigma 1,5." ¿Y esto qué me da? y me da un vector. Sí, porque sabe que yo estudié procesamiento de imágenes y sé que si quiero obtener la matriz, lo que hago es el kernel por el traspuesto, ¿sí? Y ahí obtengo esto. Y acá chequeo la ganancia y ya eh me lo da me lo da unitario. ¿Sí? Entonces, como que no lo tengo que renormalizar. Vale, 099999 la suma de todos los todos los coeficientes. Bien, eh para aplicar desenfoque gauciano lo que hago es pedirle un filtro en X, un filtro en I y usar esta función que se llama SEP filter 2D. sirve para hacer convolución con eh filtros separables. Recibe la imagen. Este parámetro de deptica es el tipo de dato. Si le pasas -1, te devuelve el resultado en el tipo de dato de entrada. O sea, en este caso 8 y permite pasarle los kernels en X y en I. Si le pasas uno solo, te aplica el mismo procesamiento a los dos.  
 

### 01:32:24

**lse posgrados:** Yo acá podría tener dos kéres gaucianos con distintos sigma para x y, por ejemplo. Ahora, como esta esto de de hacer desenfoque gauciano es algo que se usa un montón, tenemos una función que se llama Gaution Blur, que me permite aplicar la convulsión sin tener que pasar por el por el kernel, ¿sí? eh donde le paso la imagen, le paso el tamaño que yo quiero que tenga, en este caso 17\* 17 y le paso los sigmas. Acá si le paso el sigma en x, eh, me me pone el sigma i igual al x y hace la convolución. Sí, acaso lo dejé un poco más eh explícito. Después concatenamos los resultados, o sea, la imagen original con los otros dos y mostramos la imagen. Fíjense que para mostrar una imagen con supervision tengo que hacer esto nada más. Si lo quisiera hacer con Matlot LIP, tendría que poner un plt. Figure para decirle, "Che, acá viene una imagen." Después tengo que hacer la conversión a RGB, DBGR RGB. Después tengo que hacer el YM Show, eh, ¿qué venía después? le tendría que sacar los ejes si la quiero ver sin sin los ejes Xi y después tendría que poner un PLT.  
 

### 01:33:40

**lse posgrados:** Show para poder visualizarlo. Sí. Cambio acá con una línea ya lo tenemos lo tenemos resuelto y le podemos setear acá dentro todos los parámetros de de MAT Plot Lipkap. El que nos interesa es color map, ¿sí? Por si la queremos ver en, no sé, algún color map que no sea escala de grises o RGB. Pues si le pasas una imagen en escala de grises, se da cuenta que está en escala de grises y no le tenés que pasar el color map como como matotli. Bueno, parecen idéndicas. Sí. Y esta es la imagen original. Acá recreamos un poco lo de la teórica, o sea, agarré una imagen, la contaminé con ruido, en este caso ruido aleatorio, salt y le aplico un filtro de mediana. Acá igual me zarpé con el ruido. Esto ya está como casi irrecuperable en la práctica. Si tienen esto es porque hay algo que que está mal, tipo hay radiación o algo así en el aire. Fíjense que con el filtro de media pareciera que no hace no hace nada. Sí, pero la mediana lo deja casi casi impecable. Sí, porque claro, el ruido nunca va a estar en la mediana de de la ventana.  
 

### 01:34:55

**lse posgrados:** Acá hacemos el ejemplo de diferencia de gaucianas para extraer los bordes, ¿sí?, de las de las tapitas. Y así queda si le aplicamos el filtro de mediana. Sí, hasta diría que casi no tiene sentido extraer los bordes así si no aplico un filtro de mediana previamente. Y por último, acá tenemos un ejemplo de un sharp, un sharp masking, la imagen original, la imagen con K = 1, ya le sumo la máscara, ¿sí como está? Y acá con un K = a 50 podríamos zarparnos un poco, ¿si? y meterle un 150. A ver qué da. Bueno, fíjense que acá sí ya se empieza empieza a ver un poco un poco rara porque se empiezan a a saturar algunos píxeles. De hecho, acá se ven un montón de de puntos debido a errores de de de overflow, ¿sí?, que se dan porque estoy sumando un número demasiado grande. Pero bueno, jueguen, jueguen con esto después. Y acá está el ejemplo, ¿no?, de cómo generar este gráfico con eh Matotlip. Sí, tenemos que escribir todas estas líneas de código cuando con supervillón es lista de imágenes, lista de títulos, tamaño de de la grilla.  
 

### 01:36:19

**lse posgrados:** E bueno, si tiene alguna duda de esto, eh, nos queda transformada de furi y un algoritmo de detección de bordes que hace uso de filtros, pero con un par de cosas un poco más complejas. Yo saltearía lo de furiar, pasaría a lo de a lo de extracción de bordes, que está más o menos ligado a esto todavía y después por ahí metemos un intervalo después de la práctica de bordes contra forma de furiar que tiene tanto un contenido bastante extenso en la teórica y también en en la práctica. De hecho, la práctica es teórico práctica en furiar. Así que todo esto lo salteamos. Eh, parece que salía Fier Fier. Este es el TP2. Ahí está. Detector de bordes. La idea de del algoritmo de Cani, sí, es hacer detección de bordes, pero de una manera un poco más inteligente, porque si yo aplico un filtro de bordes a una imagen, voy a obtener cosas que son bordes, voy a obtener cosas que son ruido y voy a obtener cosas que podrían parecer bordes, pero en realidad es ruido o al revés. ¿Sí? Entonces, no puedo diferenciar bien qué cosa es qué cosa si simplemente aplico un filtro y ya está.  
 

### 01:37:46

**lse posgrados:** Entonces, lo que hace Cani es eh bueno, una manera un poco más compleja de extraer bordes, pero también un poco más confiable. Para arrancar lo que hacemos es eh filtrar con un kernel de desenfoque gaucial, ¿sí? O sea, hacemos una especie de extracción de ruido, lo atenuamos un poquito. Después obtenemos el gradiente tanto en X como en I. O sea, agarramos la imagen y le aplicamos un filtro de Sobel en X y un filtro de SOEL en I, formando dos imágenes. Una que tiene bordes en X, otra que tiene bordes en I. Calculo el módulo, ¿sí? obteniendo una imagen de bordes, pero ya olvidándome de la dirección o de del signo de del borde y calculo una imagen de ángulo. ¿Sí? La imagen de de módulo, creo que acá se entiende, píxel a píxel, se eleva al cuadrado, se suma, se calcula la raíz cuadrada y el arcotangente, esto también es píxel a píxel. Sí, agarro la imagen en i, el primer píxel, imagen en X, el primer píxel, dividido arcotangente y así para todos los para todos los píxeles. Entonces, de acá me quedo con dos imágenes, una imagen de módulo, una imagen de de ángulo, ¿sí?  
 

### 01:39:06

**lse posgrados:** Y el ángulo sí se va a corresponder, o sea, al píxel de a su correspondiente píxel, valga la redundancia, en la imagen de de módulo. Eh, o sea, acá ya tendría una imagen de borde, ¿sí? O sea, es lo que nosotros haríamos si quisiéramos extraer bordes de Foque auciano. Filtramos parecido a, no sé, el ejemplo de de Log de la Placiana de la Gauciana. Acá viene la parte la parte nueva que es la supresión de no máximos. La idea es ver si cada píxel de la imagen de módulo es un máximo local, ¿sí? Local dentro de una dentro de una ventana, ¿sí? Entonces, lo que se hace es para cada píxel me fijo cuál es el ángulo que le corresponde. Por ejemplo, en este caso para el píxel A le da este ángulo, ¿sí?, que está acá. Entonces, eh digo, che, este ángulo tiene que caer en alguno de estos bins, que cada bin se corresponde a la orientación de cada uno de los ocho vecinos que tiene alrededor. Entonces, este ángulo cae acá para este píxel y lo que hago es preguntarme si A es mayor que B, ¿sí? Y que C, que son como los vecinos inmediatos en la dirección del gradiente y en la dirección opuesta.  
 

### 01:40:29

**lse posgrados:** Sí. Si esto se da, lo que voy a lo que voy a hacer es eh conservar. Ah, si esto no se cumple, lo descarto. Sí, digo, "Che, este no es un un candidato a a borde." Entonces, acá pasamos de la imagen de bordes preliminar a algo así que tiene los bordes un poquito más finitos porque se fueron eliminando algunos píxeles. Ahora, acá en este proceso, el ruido seguro que se va a eliminar, pero también voy a eliminar píxeles que en realidad son píxeles de borde. Entonces, se viene el último paso, que es un umbral, pero con histerrisis. Eh, los valores, ¿sí? de histéricis son parámetros del algoritmo y son dos, o sea, tenemos un umbral alto y tenemos un umbral bajo. Y lo que hacemos es preguntarle a cada uno de los píxeles que sobrevivió en la imagen de módulo si están por encima o por debajo del umbral alto. Si están por encima, esos quedan, ¿no? No hay mucho para hacer. Eh, si están por debajo, esos también los los borro. Sí. Eh, y si están en el medio, acá lo que hago es hacer un análisis de componentes conectados y digo, "Che, ustedes que están en el medio, ¿están conectados, o sea, tienen de vecino algún píxel que esté por encima del umbral alto?" Por  
 

### 01:42:01

**lse posgrados:** ejemplo, en este caso no. Entonces todos estos los descarto. Pero en casos así donde los píxeles están conectados con píxeles del umbral alto, estos los rescato. Digo, "Che, bueno, ustedes sobreviven y son píxeles de bordes y por más que los haya eh eliminado en el caso en el paso anterior, entonces pasamos de esta imagen a esta imagen donde reviven algunos algunos píxeles y mueren otros." ¿Sí? Eh, entonces, bueno, este es el lo que se conoce como el algoritmo de de Cani y es eh uno de los algoritmos por ahí eh óptimos para para detectar bordes. Sí, acuérdense que los bordes son como el el punto de partida en realidad de algoritmos más complejos que algunos ya los vamos a empezar a ver la clase la clase que viene. Y esto de construir una matriz de ángulos, lo vamos a una imagen de ángulos. Bueno, es una matriz. Eh, lo vamos a lo vamos a ver repetido, eh, más adelante, ¿sí? Así que estudien el algoritmo. Ahora en la práctica lo vamos a lo vamos a pegar una una revisada igual, pero ténganlo presente, más allá de que el resultado del algoritmo de Canii es importante porque me da una imagen de bordes, eh por ahí incluso es igual o más importante lo todos los conceptos que que surgen de los pasos intermedios.  
 

### 01:43:23

**lse posgrados:** Sí. Eso de que yo compongo una imagen de módulo y que tengo una imagen de ángulo que se corresponde píxel a píxel. ¿Sí? Eh, bueno, veamos una práctica de esto y después seguimos conformada de de furier. La práctica es s super básica. Sí, tenemos por un lado kernelciano kernels de Sbel. Sí. Acá le indicamos qué es lo que queremos ver, eh, si queremos algo en X y nada en I o nada en X y algo y algo en I. Sí. Acá con esto pueden jugar, ¿sí? Este, lo ponen true and false y lo obtienen con con signo o binarizado. Sí. El resultado es lo mismo que vimos en en la teórica, no hay mucho para comentar acá. Eh, lo mismo, le pasan el tamaño el tamaño del kernel, el tipo de dato en el cual quieren que se devuelva el resultado. Acá hay una implementación manual de el detector de CANI, detector de bordes de CANI. No es la implementación más óptima, sí, pero si es una implementación paso a paso que más o menos se condice con lo que vimos en en la teórica, ¿sí? Que está bueno para para entenderlo de una manera fácil.  
 

### 01:44:50

**lse posgrados:** Y lo que tenemos acá, sí, es la implementación del detector de de bordes de CIA en Open CB, ¿sí? Que es esta función y recibe como parámetro. Los parámetros principales son los umbrales, pero también tenemos lo que es eh la apertur size, que creo que es el tamaño de de la matriz de bordes que van a la matriz de bordes de del kernel que va a ser la extracción preliminar de bordes y un par de un par de cositas más, eh, y este es el el resultado. Si acá simplemente está aplicado la imagen, digamos, no no es que está bien o está mal, ya los parámetros o los hiperparámetros los van tuneando ustedes a medida que, digamos, que encuentran una aplicación para esto, que de hecho la van a encontrar porque ya en el TP3 algo de esto van a tener que van a tener que usar. ¿Alguna duda o pregunta? Bueno, veamos la parte teórica de Furier, ¿sí? Y después pasamos a por algún intervalo o seguimos con la práctica. Bueno, nos habíamos quedado acá. Transformada de furiar. Bueno, la transformada de furiar nos permite obtener el contenido espectral de una imagen. Por ahí voy a ir un paso más atrás, ¿sí?  
 

### 01:46:25

**lse posgrados:** Para los que nunca vieron forma de furiar para poder entenderlo eh de una forma más simple con funciones. Sí, la imagen también es una función, pero es 2D y se complica por ahí.  
**Daniel Bazan:** M.  
**lse posgrados:** Con funciones unidimensionales es un poco más simple. Sí. Si yo tengo una señal senoidal, ¿sí?, en el dominio temporal, ¿qué es lo que caracteriza a la señal? O sea, yo tengo una señal senoidal y la quiero describir. ¿Qué qué qué le tengo que pedir? O sea, ¿qué tengo que saber para poder caracterizar una senoidal?  
**Juan Pablo Gonzalez:** y que es periódica y continua.  
**lse posgrados:** No, bueno, esos son atributos, digamos, de de muchas señales que no necesariamente son ideales, pero por ahí que la quiero la quiero eh Okay.  
**Juan Pablo Gonzalez:** Ah, sí. Lo que mencionan ahí, la fase, el periodo, la amplitud.  
**lse posgrados:** Bueno, bien. O sea, el periodo, la fase y la y la amplitud. Vamos a asumir que la fase es cero, ¿sí? Para hacerla fácil, ¿eh? Pero tenemos la amplitud, ¿sí? que me dice cuánto mide la la senoal y el periodo, sí, que me dice cuánto mide, pero en el otro eje.  
 

### 01:47:40

**lse posgrados:** Sí, a mí me gusta expresarlo un poco distinto. En lugar de usar el periodo, uso la frecuencia que es la inversa, ¿no? Es 1 sobre t, porque el periodo está bien, sí está en unidades de tiempo, pero es medio medio complicado. O sea, si yo te digo, "Che, esta señal tiene un periodo de 1 milisegundo, yo no sé, no me lo imagino, no la puedo no la puedo medir si con un cronómetro o algo así, pero si yo hablo de frecuencia, sí, la frecuencia en herz ahí es más claro porque yo sé que eh si tengo un periodo de un segundo, tendría una frecuencia de 1 herz y yo sé que es una señal que hace una oscilación Sí, por segundo. Entonces es un poco más simple. Bueno, eh la transformada de Furiar lo que hace es representar un poco distinto las señales. Sí. y una señal que tiene una amplitud de uno y una frecuencia de 1 Hz, la va a representar en el dominio de Furiar como algo que tiene amplitud uno o proporcional a uno. En realidad acá depende de la bibliografía. Sí. Eh, y algo que está ubicado en un herz. Sí. O sea, pasamos de una representación temporal porque en realidad, a ver, acá yo dibujé un periodo, pero esta señal es infinita, ¿sí?  
 

### 01:49:20

**lse posgrados:** Va para un lado y va para el otro, ¿sí? O sea, seno de de omega t en el dominio de furier se representa como un punto, ¿sí? O sea, es un punto que tiene una altura proporcional a la amplitud de la señal en el dominio temporal y está ubicado en el eje que se corresponde con la frecuencia de la señal. como que es un poco más sintética la representación. Eh, bueno, las senoidales son eh señales bastante particulares, al igual que las exponenciales y se pueden representar de esta forma, o sea, están compuestas por una única frecuencia. Pero hay otras señales, como puede ser, por ejemplo, una señal triangular o una señal cuadrada o no sé, cualquier una señal de voz hablada, ¿no? Que si nosotros la grabamos vamos a ver que es una cosa así, que no tiene una forma determinada. Yo también le puedo calcular la transforma de furiar porque, bueno, hay un teorema que dice que yo puedo representar cualquier señal periódica eh como una combinación lineal de senos y cosenos. Sí. Eh, los coeficientes de esa combinación lineal son los coeficientes de Furier eh, y están un poco relacionados a la a la transformada de espurier. Eh, entonces cualquier señal yo la puedo representar como palitos asumiendo que todas las señales están compuestas por eh funciones senoidales.  
 

### 01:50:45

**lse posgrados:** ¿Sí? Porque encima el seno y el coseno y son la misma función, pero uno está un poco desfasado de lo otro. ¿Sí? Entonces, cualquier señal yo la puedo representar con palitos, básicamente y ver su contenido espectral. Eh, entonces, bueno, para una señal un poco más compleja, ¿sí? En lugar de tener un solo palito por ahí, lo que tendría es una cosa, una cosa así, ¿no? Como el logo de de Cisco que son justamente es un espectro. Eh, bueno, obviamente para señales complejas yo la puedo escribir como una combinación casi infinita de senos y cosenos. Por eso uno ya no no está dibujando palitos, sino que directamente dibuja la envolvente del espectro, ¿sí? Y le termina quedando lo que se conoce como el diagrama de bode muchas veces. Sí. Eh, obviamente esto es en el caso en el caso discreto, ¿sí? Pero está bien porque es el caso en el que nosotros trabajamos. En imágenes lo mismo. Eh, el tema es que en imágenes la el contenido espectral o las frecuencias no solo tienen una amplitud que es proporcional a el nivel de intensidad y una frecuencia que está relacionado a cómo cambia ese nivel de intensidad, sino que también tiene una dirección porque yo me puedo mover en un plano para las imágenes.  
 

### 01:52:13

**lse posgrados:** ¿Sí? Entonces, la transformada de furier de una imagen no la hago en una una recta, sino que la tengo que hacer en 2D. Desde el punto de vista práctico, una transformada furior unidimensional o bidimensional se calcula igual. Sí. La fórmula de una transformada de fulier discreta es esta que tenemos acá. Sí, como ven, es una combinación, ¿no? De sumas y productos. Y yo las sumatorias las puedo mover, como estos son términos que se multiplican, también los puedo mover. Entonces, puedo pensar esto como dos transformadas de furier unidimensionales. Una transformada de furier corre sobre las filas y la otra corre sobre las columnas. ¿Sí? Eh, ¿para qué nos sirve esto? Bueno, primero la transformar de Fier única para cada imagen, entonces ahí no hay ambigüedades y podemos ver cosas que por ahí en el dominio espacial se nos escapa. Es un muy buen descriptor para extraer características. También sirve para acelerar los cálculos, porque acá hay algunas propiedades. La única que me interesa es esta de acá. La convolución en el dominio de Furier es un producto punto a punto, ¿sí?  
 

### 01:53:34

**lse posgrados:** Lo cual está buenísimo porque claro, es más barato multiplicar números uno con uno sola vez que hacer una suma de convoluciones. ¿Sí? O sea, obviamente es mucho más eficiente y uno podría decir, "Che, bueno, paraá, o sea, por ahí en la convolución es eficiente, pero para llegar a una transformada de furier tené que hacer esta cuenta que son dos sumatorias. Hay un montón de productos, hay unos números complejos metidos ahí, sí, pero hay un algoritmo que es el algoritmo de la FFT, que es eh FFT viene por Fast Furier Transform, que es una manera superficiente de calcular transformada de Furier y es tan eficiente que conviene aplicar transformada de Furier y después filtrar el dominio espectral. Sí. Y esto se hace en varios varios campos, no solo en imágenes. Sí. Eh, bueno, en el campo del audio hay algo que eh se, por ejemplo, el ejemplo por hoy más claro son las reverbs convolucionales. se suele hacer algo bastante interesante que es captar la función transferencia de distintos recintos, puede ser el hall de una iglesia o cosas así que tienen un tiempo de reverberación copado y después mediante la convolución de la respuesta al impulso con una señal eh recrear como si algo fuera reproducido en eh en ese en ese ambiente.  
 

### 01:55:02

**lse posgrados:** Eso se hacía antes. Hoy en día como que se ramificó un poco y uno puede crear respuestas al impulso artificiales de entornos que no se pueden construir físicamente, ¿sí? creando efectos bastante locos, pero claro, son respuestas al impulso larguísimas, entonces una convolución ahí tardaría mucho tiempo, pero en el dominio de Furier eh se hace de un saque porque es simplemente un producto punto a punto. En imágenes se da lo mismo, sí, hacer convolución es costoso. Eh entonces podría hacer transformada de furiar del gel, transformada de furiar de la imagen y eh bueno, filtrar en el dominio espectral. Acá tenemos un par de ejemplos. La transforma de Furier, ¿sí? Eh, en el dominio espectral lo que me dice es lo que está cerca del centro son bajas frecuencias. A medida que me voy alejando son altas frecuencias. Entonces, si quiero hacer un pasabajos, podría generar esto, ¿sí? que es una especie de máscara en el dominio espectral que deja presente solamente lo que está dentro de un determinado radio. ¿Sí? Eh, el radio determina la frecuencia de corte y por encima de ese radio anulo todas las frecuencias.  
 

### 01:56:22

**lse posgrados:** Y en el caso de pasaltos es al revés, anulo todo lo que está dentro de un determinado radio y por encima de esa frecuencia de corte dejo todo como está. Eh, bueno, acá tenemos algo que es un kernel gauciano, así chiquitito, insertado en una imagen y su transformada de de furier. Eh, esto es una propiedad de de la gauciana, ¿sí? La transformada de furier de una gausiana es otra es otra gausiana, ¿sí? eh y viceversa también no puede diseñar un kernel gauciano en el dominio furier, antitransformarlo y ese kernel va a existir en el dominio en el dominio espacial, ¿sí? Va a seguir siendo gaciano. Algo que no se cumple con este tipo de filtros. Sí, al trabajar en el dominio digital uno puede hacer cualquier cosa, ¿sí? Digamos porque el lenguaje de programación, el framework me lo permite. El tema es que fíjense que la transformada de Furier me devuelve números complejos. Eh, y si yo antitransformo esto, o sea, quiero ir del dominio espectral al dominio espacial, probablemente estos filtros no existan. ¿Sí? O sea, ¿cómo cómo saber si existen o no?  
 

### 01:57:34

**lse posgrados:** Y me van a dar números complejos y yo no puedo tener números complejos en el dominio espacial de de imágenes. Es medio inconcebible eso. Entonces, ¿cómo sería nuestro nuestro algoritmo? Si queremos filtrar en el dominio espectral, tenemos que hacer la transformada de furiar de la imagen. Sí. ¿Cómo lo hacemos? Bueno, está está acá. Tenemos que hacerla por filas primero, después lo hacemos por columnas y obtenemos mi transformada con números complejos. Y después, como todo número complejo, lo podemos representar por una matriz de módulo y una matriz de fase. ¿Qué es esta cosa de acá que dice transformada 2D y después central? eh por cómo está implementado el algoritmo de la FFT. Y esto pasa pero en todos los frameworks donde hay una FFT implementada, pasa en un P, pasa en Matlab, pero en todos lados. Eh, si uno calcula la transformada de furiar de una señal, eh, lo que va a obtener es la transformada de furiar de la señal, pero con ceros en los costados. Sí, esto es rarísimo, pero pero es así. Entonces, si yo tengo una señal de 1 Hz y esto es un cero, 1 Hz va a estar acá y también va a estar acá.  
 

### 01:58:51

**lse posgrados:** Sí, el espectro tiene su contraparte negativa también, o sea, negativa del otro lado de del eje, no no que sea mala. Lo lógico para nosotros es tener un cero en el medio y que los números crezcan para la derecha y decrezcan para la izquierda. Eh, pero no, eso no es así para los algoritmos de la FFT, las implementaciones de la FFT. Entonces, todos los frameworks que hacen FFT tienen una función que se llama FFT shift o algo parecido, que lo que hace es llevar el cero a un cero que nosotros podamos entender. Sí, esto solamente es para poder representar las cosas, ¿no? Para ser filtrado, esto a mí no me importa. Yo filtro, aplico la transformada, multiplico punto a punto y antitransformo y ya tengo el resultado. Pero si lo quiero graficar, lo quiero representar, eh bueno, conviene llevar el cero al origen para poder interpretar un poco los resultados de forma de forma gráfica. Por eso está como este paso acá de eh centrar el cero. Y bueno, en imágenes es más complejo todavía porque no tengo dos ceros, tengo cuatro orígenes. Sí, cada origen está acá. Entonces, interpretar una transformada de furial en 2D y es medio un kilombo.  
 

### 02:00:11

**lse posgrados:** Por eso acá sí eh conviene hacer el FFT shift porque bueno, mal que mal en una dimensión te das cuenta y decís, "Bueno, tenés el cero acá, contas los números al revés y listo." Pero en una imagen ya se se complica un poquito. Bueno, eh yo deía de meter un intervalo, ¿sí? Y seguimos con la parte teórico práctica de de esto. Sí. La idea es ilustrar un poco más esto de la transformada de Fulier en señales para que se entienda cómo es el pasaje. ¿Sí? Vamos a ver señales de alta frecuencia, señales de baja frecuencia para ver un poco cómo es esa representación, porque que la señal sea de alta frecuencia o baja frecuencia, por ahí en el dominio espacial yo lo puedo ver porque claro, entran más eh oscilaciones en el mismo periodo, entonces es una sea de alta frecuencia, pero bueno, eh creo que hasta que no no tengamos una práctica de esto, no se termina de de comprender. Son 9 y cu. Vamos intervalo hasta las 9 y 9:20. Y después cerramos con la teórica y la explicación del TPM. Bueno, le metemos unos minutos más, ¿sí? Y ya con esta con esta práctica cerramos.  
 

### 02:18:29

**lse posgrados:** Acá tenemos algo que ya creo que fui comentando. Sí, tenemos una una señal unidimensional y su contraparte en el dominio espectral. Eh, esta misma señal yo la puedo tener en una imagen también, ¿sí? en un plano donde eh los picos sí, bueno, van a ser eh máximos o mínimos de eh luminancia, máximos y mínimos de de intensidad, si uno quiere ver en escala de grises, ¿sí? O sea, esto, ¿qué sería ahí? Sería como una especie de degradé que va oscilando, o sea, arranca en negro y va subiendo hasta que se vuelve blanco y después empieza a bajar de vuelta hasta llegar a negro y después vuelve a subir. Sí. Eh, y la transforma de furiero es idéntica, solamente que la voy a ver en un plano, porque no es lo mismo tener esta senoidal así horizontal que tenerla, no sé, con un ángulo de 45 gr, por ejemplo. Por eso es que eh voy a tener misma ubicación, misma amplitud, pero distinto, o sea, misma ubicación, eh, pensándola como distancia al origen, ¿no? Pero puede tener distinto ángulo respecto de la la horizontal o la vertical o como lo querramos. medir.  
 

### 02:19:50

**lse posgrados:** Entonces, yo acá lo que hice fue cargar dos imágenes, una de alta frecuencia y otra de baja frecuencia. Ahora las vamos a las vamos a graficar. Calculé su transformada de furier acá con Numpie. Apliqué FFT shift para centrar esa transformada de furier. Y esto es solamente para eh ajustar un poco la visualización, porque algo que por ahí no les comenté, eh cuando uno trabaja con con frecuencias eh ¿qué es lo que tiene? Y puede estar esto en señales, ¿no? Podría estar trabajando con en un mismo gráfico señales que van dentro de la gama de los herz. O sea, tengo un eje que tiene que representar unidades y tiene que representar millones. Entonces, uno, y lo mismo sucede con amplitud, ¿sí? Uno puede tener componentes que son muy pequeñas y componentes que, bueno, tienen una amplitud capaz comparable a la amplitud de la señal original. Entonces, uno suele graficar el eje de la frecuencia en décadas o en octavas, ¿sí? y el eje de la amplitud eh como un eje logarítmico simplemente para poder apreciar todas las los distintos niveles de amplitud o de intensidad en este caso. Eh, bueno, y lo que hago es extraer los perfiles, un perfil de cada de cada imagen.  
 

### 02:21:20

**lse posgrados:** Ahora lo graficamos. Y dónde dónde se me fueran los los gráficos acá. Imagen de baja frecuencia, imagen de alta frecuencia. Estos son los dos perfiles. Fíjense que claro, para un mismo periodo, acá extra para un mismo longitud, ¿sí? extraje píxeles. En la imagen de alta frecuencia entran más periodos, justamente porque la frecuencia eh es más alta y en la imagen de baja frecuencia entran menos periodos porque, bueno, la frecuencia es es más chica porque algo es de baja frecuencia. Y tenemos acá la transformada de furier que también está a escala. Entonces vemos que eh las componentes de frecuencia están más lejos en el centro, más lejos del centro, perdón, en la transformada de alta frecuencia que eh en la transformada de baja, que está más cerca de del centro. Sí, porque cuanto más cerca del centro estoy, más baja es la frecuencia. Cuanto más alejado, más alta es. Ahora, esto tiene las componentes principales, ¿sí? Pero también tiene como máximos de intensidad eh en otras partes de la imagen. Sí. De hecho, acá por ejemplo se ve en la imagen de la izquierda como como si fueran fueran réplicas, ¿no?  
 

### 02:22:46

**lse posgrados:** Y están como equpaciadas un poco y hasta que hasta que desaparecen. Bueno, eh son armónicos. Sí, en general es casi imposible lograr una señal que esté compuesta de una única frecuencia. Entonces, lo que tenemos es la frecuencia fundamental y los múltiplos de esa frecuencia, además de cualquier otro otro ruido presente. Fíjense que esto no es una senoidal perfecta, o sea, acá como que hay cosa medio abrupta que también se modela como componentes adicionales en su transformada furior. Si no, veríamos todo negro y dos puntitos a los costados y y nada más. ¿Se entendió esto? ¿Quieren que repita algo? ¿Tienen alguna consulta? Sí, Nael.  
**Nahuel Otonelo:** Hola, ¿se escucha bien?  
**lse posgrados:** Sí, sí.  
**Nahuel Otonelo:** Ah, buenísimo. Yo no entendí la relación entre la sinusodial que dibujaste y la imagen. Esa parte me perdí.  
**lse posgrados:** Eh, la las las senoidales son perfiles que extraje la imagen nada más, ¿eh?  
**Nahuel Otonelo:** Ah.  
**lse posgrados:** O sea, la imagen. A ver, ¿dónde dónde está?  
 

### 02:23:59

**lse posgrados:** De acá. FFT LF. Sí. O sea, la imagen tiene esto, tiene franjas. Bueno, por ahí conviene verlo acá en realidad, sí, que forman, o sea, son franjas que en el medio tienen un degradé, ¿no? Esto va de oscuridad a de 0 a 255 y después sube. Yo lo que hice para ver la senoidal es extraer eh una una fila, la fila 50. Eh, entre estos valores, sí, o sea, traé trasé un perfil de de la imagen nada más, o sea, me paré en la fila 50 y dije, "Che, entre el píxel 10 y el 60, 50 valores, dame dame los todos los valores." Simplemente eso, o sea, un perfil eh para de cuál de cuál de las tres.  
**Nahuel Otonelo:** Está bien. Y cómo lo y digamos en los ejes, ¿qué sería eh digamos el eje, el eje X de la imagen esa? No, no, de la de las que son tipo temporales, de las que parecen temporales.  
**lse posgrados:** Bien, eh, en el eje X tengo los 50 píxeles que extraje. O sea, es un índice de píxeles porque saqué píxeles que van de del 10 al 60, que son 50 valores.  
 

### 02:25:20

**lse posgrados:** Y en el eje I lo que tenés es el nivel de intensidad, o sea, es el brillo.  
**Nahuel Otonelo:** Ah, okay, está bien, está bien.  
**lse posgrados:** O sea, es como yo me paré en el en la fila 50 de la imagen y me llevé 50 píxeles.  
**Nahuel Otonelo:** Entendí. Sí, sí. O sea, estás estás graficando el valor de el escala de grises de de 0 a 255, que bueno, cuando es negro es eh cero.  
**lse posgrados:** Exactamente.  
**Nahuel Otonelo:** Está bien, ahí entendí. Está bien.  
**lse posgrados:** Es eso.  
**Nahuel Otonelo:** Y una cosa te quería preguntar, eh, si está mal pensar que que esa fila que vos dibujaste es como la transformada de Fier de de solamente esa fila, es Ah, okay.  
**lse posgrados:** No, no, la fila que dibujé es la fila. No, no, no, no, no.  
**Nahuel Otonelo:** No, no tiene nada que ver una cosa con la otra, entonces.  
**lse posgrados:** Simplemente en lugar de graficar toda la imagen, grafiqué una fila.  
**Nahuel Otonelo:** Está bien, gracias.  
**lse posgrados:** Bien, bueno, como a lo largo del notebook vamos a usar mucho esto de aplicar FFT, hacer FFT shift y tomar 20 logaritmos, directamente me definía acá una función, ¿sí?, que se llama FFT mod, que lo que hace es  
 

### 02:26:36

**lse posgrados:** calcula todo eso. Sí, Daniel.  
**Daniel Bazan:** Perdón. Ahí continuando lo que te estaban preguntando, la primera imagen esa tridimensional, esa sí sería toda la imagen completa, no solo una fila, ¿no? La primera que estaba arriba, ¿no? No, pero no, pero de ahí en el en el más arriba arriba.  
**lse posgrados:** Sí, son el plot de de imagen, básicamente, no no es otra cosa.  
**Daniel Bazan:** No, no, la otra no. Esas si no había otra más arriba ahí. Ese sería si la de la izquierda.  
**lse posgrados:** Esto acá.  
**Daniel Bazan:** Esa sería toda la la imagen completa, ¿no?  
**lse posgrados:** Claro.  
**Daniel Bazan:** Ya no una fila, sino todo completo.  
**lse posgrados:** Sí. Bueno, acá me definí un par de funciones auxiliares.  
**Daniel Bazan:** Gracias. Okay.  
**lse posgrados:** Una que es la que me toma el módulo, hace el FT shift y toma el 20 logaritmo para para poder graficar. Acá tengo dos funciones que me armé para definir los filtros, ¿sí?, que ahora los vamos a ver, un pas altos y un y un pasabajos, ¿sí? Que los hago a partir de esta funcioncita que define una forma circular.  
 

### 02:27:53

**lse posgrados:** Entonces, en el pasaltos, eh, el círculo va a estar negro en el centro, en el pasabajos al revés, va a estar negro alrededor y blanco en el centro para poder anular las frecuencias de interés. Vamos a trabajar con una imagen un poquito más eh un poquito más compleja, ¿sí? Y es esta de acá. Y esta es la transformada de Furiar. Fíjense que ya en una imagen real eh es difícil decir, "Che, bueno, eh este es el componente que se corresponde con este patrón porque tenés un montón de cosas y algunas ni siquiera tienen pinta de ser periódicas. Entonces, nada, es un rejunte de píxeles con distintos niveles de intensidad. Sí, la idea es aplicar un pas altos y un pasabajos a esta imagen, pero en el dominio espectral. Acá defino los filtros con las funciones de más arriba, ¿sí? Y esta es su representación, digamos. No, no hay mucho más para comentar. Es lo que estaba en la teórica, ¿sí? Tenemos el pasajos y tenemos el pasos. Para aplicar los filtros a la imagen. Bueno, necesito aplicar la transformada de Furier a la imagen y tengo que hacer FFT shift porque los filtros ya los definícto respecto de del centro.  
 

### 02:29:13

**lse posgrados:** Sí, esto solamente es necesario para este caso. Si yo tuviera los filtros eh definidos de otra forma, eh tal vez no tendría que aplicar FFTs shift. Eh, el filtrado acá es multiplicar punto a punto las dos transformadas y después eh aplicar la transformada de Fulier inversa. Sí, es bastante bastante simple. Eh, ahora si bien la imagen partió del mundo real y cuando el antitransformo debería tener coeficientes reales, eh esto no se da, hay un error numérico ahí. Eh, si uno hace un print de los valores, va a ver que tiene un número real más algo imaginario por e a la men 12, que nada es cero, pero está presente. Y para poder graficarlo le tomo la parte la parte real. Sí, simplemente por eso. Eh, bueno, y graficamos lo que lo que nos da. Bueno, en el caso del pasabajos, uno esperaría ver la imagen difuminada y está difuminada. Y en el caso del pasaltos, eh, bueno, tenemos una imagen que tiene los bordes un poco más eh resaltados, pero se ve se ve un poco rara. Sí. O sea, esto no se ve como un una imagen filtrada con, no sé, un kernel gauciano o con el kernel ese que tenía todo unos.  
 

### 02:30:49

**lse posgrados:** Sí, se se ve raro. De hecho, se ven como una especie de ecos de la imagen, como una especie de réplicas acá y son fenómenos que no se dan en en una convolución cuando filtramos en el dominio espacial. Eh, bueno, esto, ¿por qué se da? Lo adelanté un poco en la teórica, nosotros diseñamos los filtros en el dominio espectral de una sin saber si eran filtros que podían existir en el dominio espacial. Sí. Acá en imágenes para que el filtro exista, la antitransformada tiene que ser real, ¿sí? Tiene que ser números reales. Eh, si nosotros antitransformamos los kernel que que hicimos, no van a dar números reales y no es como acá que nada, hay un pequeño residuo y lo y le tomo el reel de un país para quedarme con el valor final. Va a ser un valor grande, o sea, la componente imaginaria va a ser muy grande ahí. Entonces, ¿qué tenemos que hacer? Y si aplicamos eh filtrado en el dominio espectral, siempre tenemos que partir de un kernel que exista en el dominio espacial. Por ejemplo, el kernel gauciano, pero yo no lo puedo transformar de una porque un kernel gauciano es un kernel de 3x 3, de 5\* 5, de 7\* 7. La imagen capaz está en 4K, ¿eh?  
 

### 02:32:05

**lse posgrados:** y no puedo hacer una multiplicación punto a punto de algo que está en 3\* 3 con algo que está en 1280 por, no sé, por 1024 por tirar una resolución. Entonces yo tengo acá esta función auxiliar que hace lo siguiente, recibe las dimensiones de la imagen porque la única forma de multiplicar algo punto a punto es que tenga las mismas dimensiones. Entonces, yo recibo las dimensiones de la imagen y recibo, bueno, los parámetros que definen el kernel. Me creo acá una imagen vacía de ceros que tiene el tamaño de la imagen original. Me creo un kernel gauciano y meto ese kernel gauciano que es un kernel chiquitito, en el centro de esa imagen de ceros. Para los que laboraron contra forma de Furier y FFT, esto les tiene que sonar porque cuando uno tenía que hacer una convolución de dos señales que tenían distinto largo, lo que hacía era completar con ceros. Sí. Era medio obvio porque completabas con ceros en una dimensión. Acá tenemos que completar con ceros en las dos para poder convolucionar. Eh, entonces completamos con ceros, o sea, insertamos nuestro kernel en el centro de algo que tiene ceros en todos lados, salgo en el centro, que es donde metimos el kernel.  
 

### 02:33:23

**lse posgrados:** Y ahí sí aplicamos la transformada de furrier. Y acá devuelvo el kernel. Y la transformada del kernel. Sí, devuelvo a los dos solamente porque los quiero los quiero graficar uno al lado de del otro. Entonces tenemos acá nuestro kernel s super chiquitito y la transformada de furier de del kernel. Y ahora sí, yo agarro esto y lo multiplico punto a punto con la imagen y debería obtener exactamente el mismo resultado que yo obtendría con una convolución o con ganan blur de de Open Cever. Bueno, lo hacemos acá. Sí, de vuelta agarro la imagen, transformada de furier, multiplico y fft shift y le tomo la parte real para antitransformar. Y este es el resultado. Sí, tenemos la misma imagen, pero ligeramente eh difuminada. Eh, dudas, consultas. Esto yo sé que es medio complicado el tema, repasen la clase después eh jueguen un poco con los parámetros, traten de filtrar sus imágenes. Sí. Eh, el ODN del algoritmo decís, "Yo creo creo que sí está en la teórica en una parte. Claro,  
**Jose Luis Diaz:** era nlog n, ¿no?  
 

### 02:34:38

**Jose Luis Diaz:** La aplicación en el campo espectral que la transformada esa, o sea, que ten una ganancia copada de hacerlo ahí bastante más rápido.  
**lse posgrados:** el es bastante. Sí, es mucho más rápido.  
**Jose Luis Diaz:** Qué buena onda.  
**lse posgrados:** Bueno, eh, hay una notebook más acá, ¿sí? Eh, que se llama videos. Esta es para que la miren ustedes porque van a tener que hacer algo de procesamiento de de video para el TP. Es s simple, digamos. eh un video, una imagen para nosotros es lo mismo. Simplemente instanciamos un handler que nos permite leer un video de a partir de un archivo. Eh, hacemos handler. Y esto nos da una variable de control y un frame. Y el frame es la imagen, la misma imagen que uno lee con IM read, por ejemplo. Sí. Bueno, acá hay un par de ejemplos de cómo leer, cómo procesar y todo eso, pero bueno, no es muy distinta a lo que ya venimos haciendo, solamente que se agrega el handler y acá un par de un par de controles en la lectura de de de los frames, porque en algún momento los frames medio que se acaban y tengo que ir preguntando si se acabaron o no.  
 

### 02:36:04

**lse posgrados:** Sí, es muy muy simple. En cuanto al trabajo práctico, bueno, la idea es agarrar un video, ¿sí? Que ya esto está subido todo en el repositorio y implementar un algoritmo de detección de enfoque, ¿sí? Algo similar a lo que tendría una una cámara, ¿sí? Cuando uno gira ahí el objetivo y te marca con verde que estás estás enfocado. Sí. Eh, la idea es agarrar este paper que también está subido al al repositorio, ¿sí? Y tiene una parte ahí donde explica el el algoritmo, donde explican la métrica que se usa. Sí, está casi al inicio. Van a ver que son cuatro o cinco pasos basados en en transformada de furier. La idea es calcular la métrica para todo el frame. Sí. y después repetirlo, pero eh calculándolo para una un rectángulo, una región centrada en el frame, pero que tiene un área del 5, el 10% del frame total, ¿sí? O sea, tengo mi el tamaño de mi imagen, el íem uno es calcular la métrica para todo el frame, el ítem dos calcularla para una región chiquitita centrada acá. Sí, como opcional pueden agarrar y definir algo que llama una matriz de enfoque.  
 

### 02:37:34

**lse posgrados:** Ahora lo voy a mostrar en el en la diapo siguiente y calcular la métrica para para esa matriz. ¿Qué es lo que se busca? Bueno, ustedes en el vídeo van a vender un montón de frames, entonces tienen que hacer un gráfico. Dice, "Bueno, frame uno, calculo la métrica, le va a dar un valor. Frame dos, calculo la métrica, les va a dar un valor. Frame 3, calculen la métrica, les va a dar un valor. Y bueno, a medida que que la imagen se va enfocando, lo que van a ver es que esa métrica va a ir subiendo. ¿Sí? La idea es encontrar el punto de máximo enfoque porque después esto queda enfocado por unos frames y después empieza a bajar. Sí, porque se desenfoca. La idea es ubicar esta esta región. Eh, la idea es para cada para cada experimento, o sea, el uno, el dos y si hacen este opcional, es generar estas estas curvas, ¿sí?, que muestren el valor de de la métrica en función de de los frames. Sí. eh y detectar, ¿no?, la la región de máximo enfoque.  
 

### 02:38:46

**lse posgrados:** Esto para la métrica que se explica en este paper. Después van a este otro paper, que también está subido, al apéndice del paper, que tiene una colección de métricas de enfoque. Ahí eligen la que más le guste o la que le parezca más fácil de implementar y repiten el proceso. O sea, ya van a tener codeadas todas las funciones para graficar y todo. Lo único que tienen que hacer es cambiar la función que usan para calcular la métrica. Por ahí lo único eh que es importante, sí, es que el algoritmo que hagan para detectar la región de enfoque la tienen que hacer de forma automática. Si no, pueden graficar la curva y decir, bueno, vi la curva y selecciona este punto y este punto. Sí, digamos, piensen ahí qué pueden hacer. Pues pueden tomar el máximo y decir, "Bueno, es este, pero está mal porque tienen que indicar una región." Podrían tomar, no sé, cuando la recta cambie de de pendiente acá, digamos, eso queda a gusto de de ustedes. La idea es que pueden indicar en el gráfico cuál es la región de máximo de máximo enfoque. Eh, bueno, acá esto es un ejemplo real de cómo quedaría la curva.  
 

### 02:39:59

**lse posgrados:** Sí, en este caso deberían decir, "Che, entre acá y acá esto está enfocado." Esto es un ejemplo de la matriz de enfoque, que es opcional el que lo quiere hacer. Para la matriz de enfoque lo que harían es calcular la métrica para cada uno de los cuadraditos en cada frame y promediar. Entonces, promedian la métrica en cada uno de cada uno de los cuadraditos y de ahí sacan un punto para un para un frame. Bueno, acá como punto extra, ¿sí? La idea es ver porque acá sabemos que la imagen está desenfocada, ¿sí? Acá podemos ver el ejemplo. Sabemos y sabemos que acá está recontraenfocada, pero va a haber una región por acá y por acá que está ahí, está como en el medio. Y esa región es una buena región donde uno podría aplicar un sharking. Sí, ahí es donde un shar masking funciona y te puede te puede aumentar un poquito esta región. Ahora, ¿dónde no tiene sentido aplicarlo? Y no tiene sentido aplicarlo acá, no tiene sentido aplicarlo acá. Y no tiene sentido aplicarlo cuando la imagen está enfocada. Sí, porque ahí incluso capaz la saturas.  
 

### 02:41:08

**lse posgrados:** Pero sí tiene sentido aplicarlo por acá y aplicarlo por acá. Sí. Entonces, la idea también esto es opcional, ver cómo hacen para mejorar un poco eh la región de enfoque con un sharking. Eh, bueno, eso eso es todo el todo el TP, ¿sí? eh para leer el video y todo eso. Tiene la notebook de de video que está todo subido y si no, bueno, me pueden mandar un mail y les respondo. Eh, vi que ya unos cuantos estuvieran haciendo entregas, todavía no miré nada, pero bueno, este fin voy a tratar de de empezar a a corregir, así no se acumulan las las cosas. preguntas del trabajo práctico.  
**fermin rodriguez:** Sí. Eh, yo no recuerdo si se entregaba por campus o si te mando el repo por mail o cómo era eso.  
**lse posgrados:** No, lo subís a un repositorio y me mand mandas el mail con el link.  
**fermin rodriguez:** Te mando el link por mail.  
**lse posgrados:** Sí, la fecha límite de todos los TPS es la clase la clase 7. Todos se van a hacer  
**fermin rodriguez:** Dale. Listo. Hago eso entonces. Gracias. Yeah.  
**Juan Manuel Calabia:** Este este TP nuevo, ¿para cuándo es la fecha límite de entrega?  
 

### 02:42:15

**Juan Manuel Calabia:** Ah, okay. Yo pensaba que era la clase que venía. Y este se puede hacer en grupo.  
**fermin rodriguez:** Son tres en total, perdón.  
**lse posgrados:** grupo, así que no con eso no hay más.  
**Juan Manuel Calabia:** Ah, okay, gracias.  
**fermin rodriguez:** Son tres en total, ¿no? O, gracias.  
**lse posgrados:** Sí, sí. Y de hecho el el último TP lo explico la clase que viene, eh, y en general, digamos, se puede resolver con los temas que vimos hasta ahora, incluyendo la clase que viene, pero yo esperaría una clase más, eh, porque el tema de la clase, qué sería la clase cinco, eh, les puede les puede ayudar un poquito para para el TP3. Pero bueno, eh nada, yo me quedo a responder consultas, eh, y si no, nos vemos nos vemos la la clase que viene.  
**Daniel Bazan:** Bueno, te consulto de nuevo lo del grupo.  
**lse posgrados:** Nos vemos.  
**Sebastian Biagiola:** Hasta la próxima. Nos vemos.  
**Daniel Bazan:** Dos de los tres TP es en grupo. Elegimos nosotros o cómo Ah, okay.  
**lse posgrados:** No pueden hacer todo como si quieren.  
**Daniel Bazan:** No entendí más última parte, entonces.  
**lse posgrados:** No, no, todos los TP son en grupo, o sea, pueden hacerlos solos, pueden hacerlos en grupo, eso hagan como prefieran ustedes.  
**Daniel Bazan:** Bien, bien. Gracias.  
**lse posgrados:** Igualmente, nos vemos.  
**Juan Manuel Calabia:** Ah, y después, ¿qué habías dicho? Al final de la materia había un examen, algo así. Ah, okay. Listo. Gracias. Hasta luego.  
 

### La transcripción finalizó después de 03:03:03

*Esta transcripción editable se ha generado por ordenador y puede contener errores. Los usuarios también pueden cambiar el texto después de que se haya generado.*