# Transformada synchrosqueezed:

La Transformada Synchrosqueezed es una técnica avanzada de análisis de tiempo-frecuencia que se utiliza para descomponer una señal en sus componentes de tiempo-frecuencia. Esta técnica es particularmente útil cuando se necesita una representación de alta resolución de la señal en el dominio de tiempo-frecuencia.

Se basa en la Transformada Wavelet Continua, pero añade un paso adicional de "sincronización" que permite una mejor localización de las componentes de tiempo-frecuencia. Esto se logra reasignando la energía de la Transformada Wavelet Continua a su centro de gravedad en el dominio de tiempo-frecuencia.

Los principales usos incluyen el análisis de señales no estacionarias, la detección de eventos transitorios y la caracterización de la modulación de frecuencia instantánea. Esta técnica se ha aplicado en una variedad de campos, incluyendo el procesamiento de señales, la física, la ingeniería y la medicina.

En este proyecto analizamos tres imágenes diferentes para las cuales usamos diferentes tipos de Wavelets para analizar cómo se comporta la SST con cada una de ellas.

Los pasos utilizados para aplicar la transformada en las imágenes son los siguientes:

* Convertir la imagen a una señal analítica bidimensional.
* Aplicar la transformada de wavelet continua (CWT) a la señal analítica usando una wavelet madre adecuada.
* Estimar la frecuencia instantánea de cada componente modulado de la señal usando la fase de la CWT.
* Reasignar los coeficientes de la CWT según la frecuencia instantánea estimada, obteniendo la transformada de synchrosqueezed (SSWT).
* Reconstruir la imagen a partir de los coeficientes de la SSWT usando la fórmula de inversión.

En el archivo 2d\_signal.py se encuentra el código con el cual analizamos 3 imágenes diferentes con la transformada de synchrosqeezed.

A continuación, le describo paso a paso lo que hace el código:

* Primero, se importan las librerías necesarias: matplotlib.pyplot para la visualización de las imágenes, numpy para el manejo de matrices y ssqueezepy para la implementación de la SSWT y su inversa.
* Luego, se define una función auxiliar llamada viz, que recibe como argumentos una matriz y un título, y muestra la matriz como una imagen en escala de grises con una barra de color y el título correspondiente.
* Después, se define la función principal llamada apply\_cwt, que recibe como argumentos el nombre de una imagen y el nombre de una wavelet madre. Esta función realiza los siguientes pasos:
  + Lee la imagen usando la función plt.imread y la convierte a escala de grises usando la función np.mean sobre el tercer eje de la matriz de la imagen, que corresponde al canal de color.
  + Aplica la SSWT a la imagen seleccionada y su resultado se almacena en una matriz tridimensional.
  + Itera sobre cada fila de la matriz tridimensional y aplica la función issq\_cwt, que devuelve la inversa de la SSWT de la fila, usando la misma wavelet madre que se usó para la transformación directa.
  + Muestra la primera capa de la matriz tridimensional resultante de la SSWT usando la función imshow, que se encarga de ajustar los ejes y los colores adecuadamente, y le asigna el título ‘Synchrosqueezed CWT’. Se visualiza la primera capa debido a que la función imshow no puede manejar funciones tridimensionales
  + Muestra la imagen original usando la función viz, y le asigna el título ‘Original Image’.
  + Muestra la matriz bidimensional resultante de la inversa de la SSWT usando la función viz, y le asigna el título ‘Image After Applying Inverse Transform’.
* Finalmente, se llama a la función apply\_cwt con tres ejemplos de imágenes y wavelets diferentes: una flor con la wavelet de Morlet, un avión con la wavelet de Bump y un oso polar con la wavelet de GMW. Estas wavelets son algunas de las opciones que ofrece la librería ssqueezepy, y se pueden elegir según el tipo y la complejidad de la señal o la imagen que se quiere analizar. La wavelet de Morlet y la de Bump son adecuadas para señales o imágenes bidimensionales, mientras que la wavelet de GMW está diseñada para señales unidimensionales.

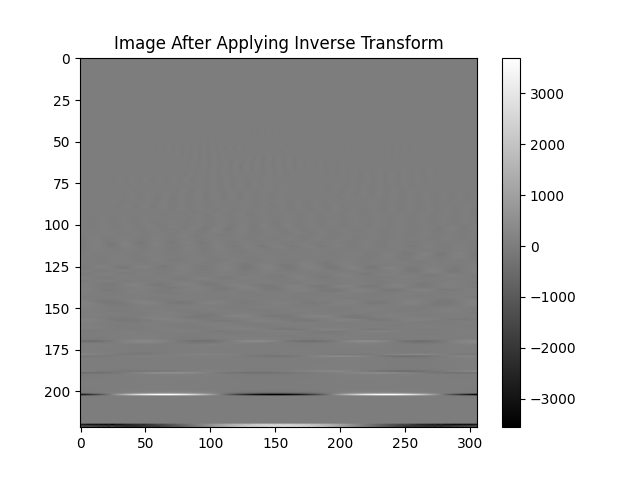
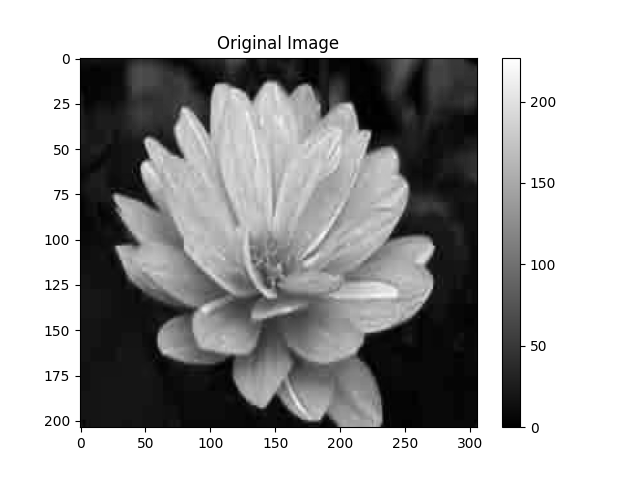
# Resultados del experimento:

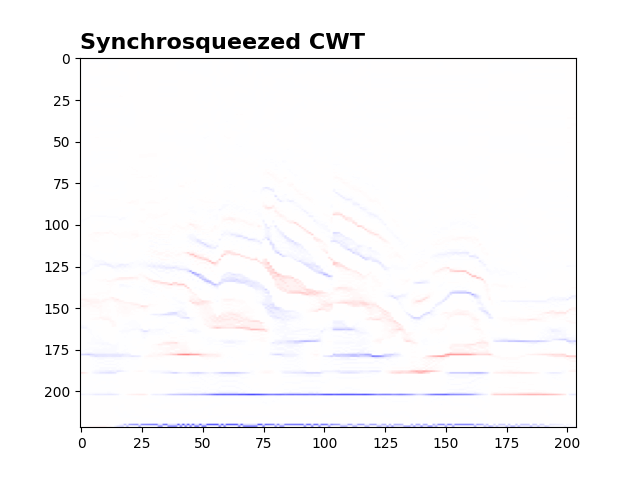
En ninguna de las fotos se pudo obtener la misma foto resultante de aplicar la inversa de la transformada. Se propone modificar los parámetros de la función ssq\_cwt para ver si se puede obtener un mejor resultado.

Además, fue posible observar que al utilizar la wavelet de Morlet como wavelet madre se obtenia un error de desbordamiento de valores de escala al tratar de dividir entre números cercanos a 0.

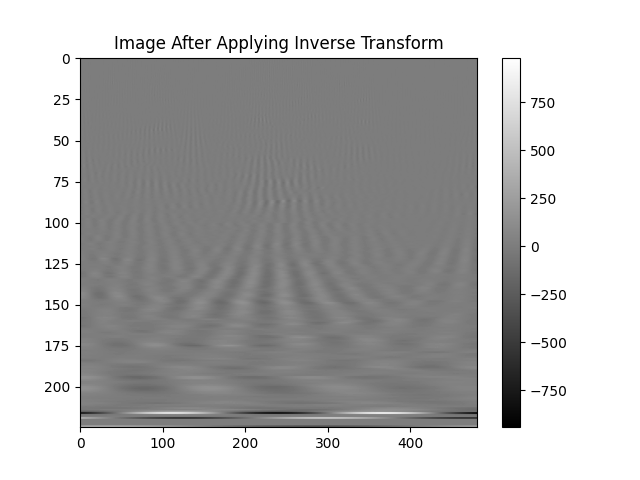
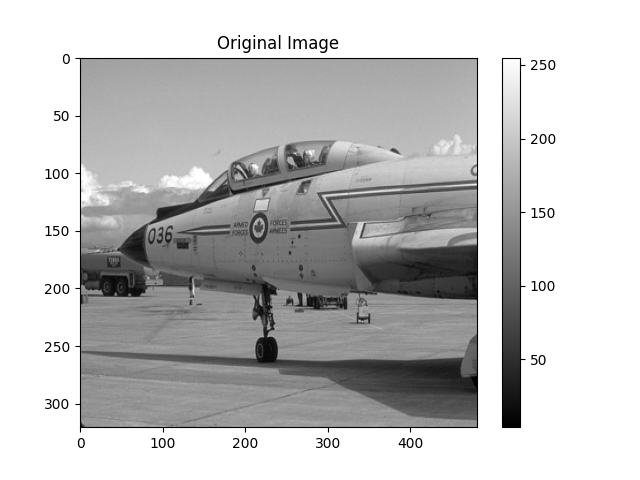
Ahora presentaremos las fotos obtenidas del experimento, las cuales se pueden obtener al ejecutar el archivo 2d\_signals.py:

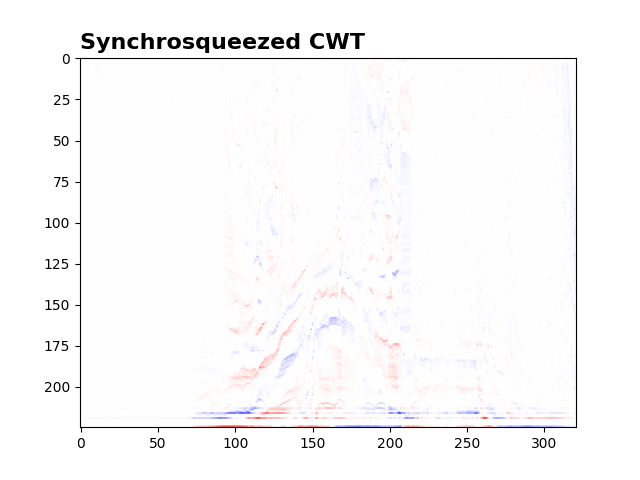
**Flower:**



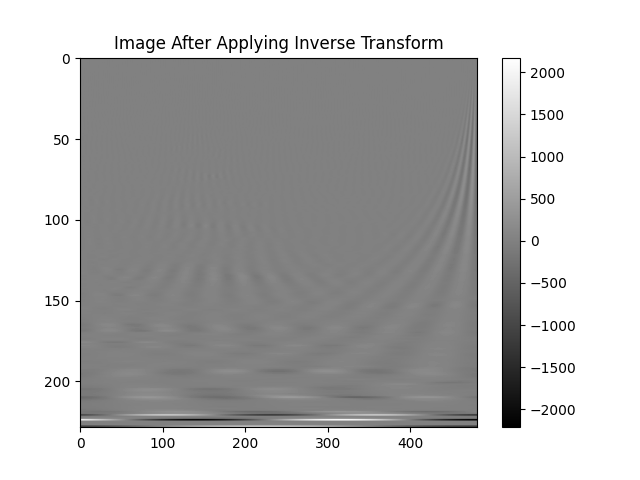
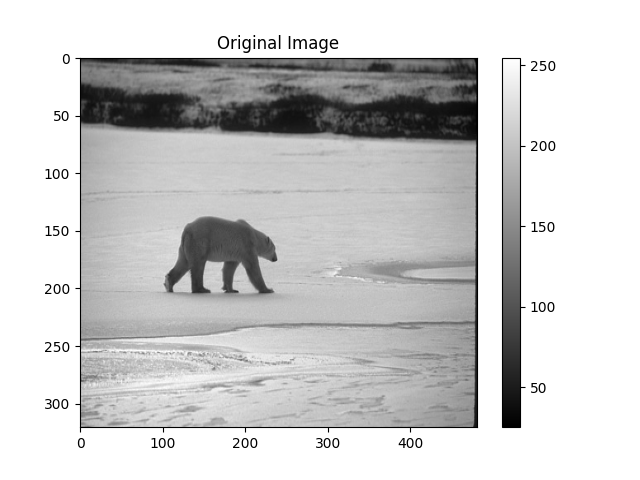


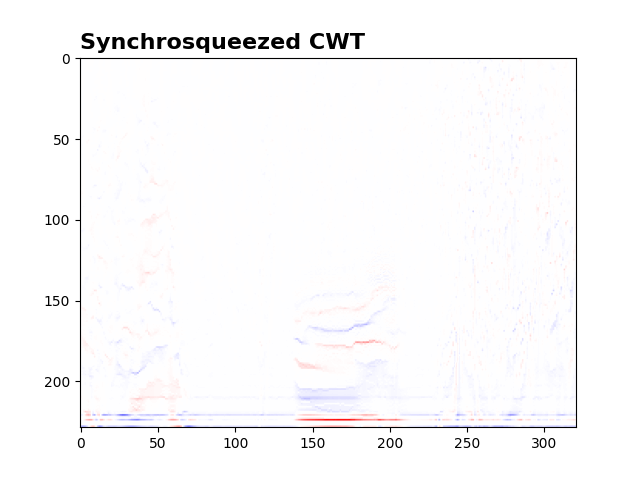
**Airplane:**

****

****

**Beard:**

****

****

# Bibliografia:

Daubechies, I., Lu, J., & Wu, H. T. (2011). Synchrosqueezed wavelet transforms: An empirical mode decomposition-like tool. Applied and Computational Harmonic Analysis, 30(2), 243–261. <https://doi.org/10.1016/J.ACHA.2010.08.002>

Daubechies, I., & Maes, S. (2017). A Nonlinear Squeezing of the Continuous Wavelet Transform Based on Auditory Nerve Models. In WAVELETS in Medicine and Biology (pp. 527–546). Routledge. <https://doi.org/10.1201/9780203734032-20>

[OverLordGoldDragon/ssqueezepy: Synchrosqueezing, wavelet transforms, and time-frequency analysis in Python (github.com)](https://github.com/OverLordGoldDragon/ssqueezepy)