Trabajando con AutoEncoders y Conceptos de Redes Adversarias Generativas

Diego Iván Perea Montealegre - 2185751

Henry Carmona Collazos - 2185965

Daniel Alejandro Tobar Alvarez - 2185884

Brahyan Camilo Marulanda Muñoz - 2185962

Facultad de Ingeniería, Universidad Autónoma de Occidente

Cali, Valle del Cauca

1. Realice una investigación de dos posibles aplicaciones donde se usen Variational Autoencoders (VAE) para la generación de algún tipo información.

Aplicación N°l: Detección de daño en una placa compuesta utilizando Variational AutoEncoders (VAE)

Para esta aplicación, se han analizado placas compuestas por una unión entre dos capas de fibra de carbono, con un núcleo Nomex tipo panal de abeja y una resina epóxica como adhesivo entre las partes. Las denominadas tipo sándwich tienen buenas propiedades mecánicas a un muy bajo peso. Sin embargo, por temas de manufactura poseen muchas veces defectos denominados delaminación. La delaminación se define como la perdida de adherencia entre el núcleo y las capas. Existe una gran dificultad en dar cuenta cuando existe esta falla, debido a que normalmente no es visualmente inspeccionable. De forma que, se realiza la detección de un dato anómalo, a partir del error generado por un VAE entrenado con imágenes normales, que se espera que no responda de forma adecuada ante imágenes con anomalías.

En el documento se investiga una forma de detectar el daño utilizando un Variational AutoEncoder cuyas entradas son imágenes de índices de daño sobre las superficies de las placas. Se obtienen mediante un método de elementos finitos basado en la curvatura de los modos de vibración. Para el entrenamiento, se incluyen las siguientes clases en el conjunto de datos:

- Clase 0: Sin daño
- Clase 1: Daño de delaminación entre 0 - 0.05
- Clase 2: Daño de delaminación entre 0.05 - 0.1
- Clase 3: Daño de delaminación entre 0.1 - 0.15
- Clase 4: Daño de delaminación entre 0.15 - 0.2
- Clase 5: Daño de delaminación entre 0.2 - 0.25

Además de las generadas computacionalmente, se cuentan con imágenes de 5 placas reales de las cuales una no posee daño, una es una clase 2, dos son de la clase 3 y una de la clase 4. Las imágenes como se observa en la figura 1 están en escala de grises y cada pixel tiene un valor real entre 0 y 1. Las imágenes tienen un tamaño de 51 pixeles de ancho y 71 pixeles de largo.

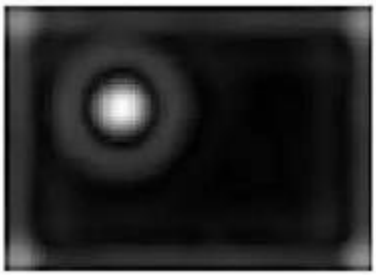


Figura l. Imagen con daño de delaminación.

A continuación, entre la figura 2 y la figura 7 se muestran tres ejemplos de cada clase.

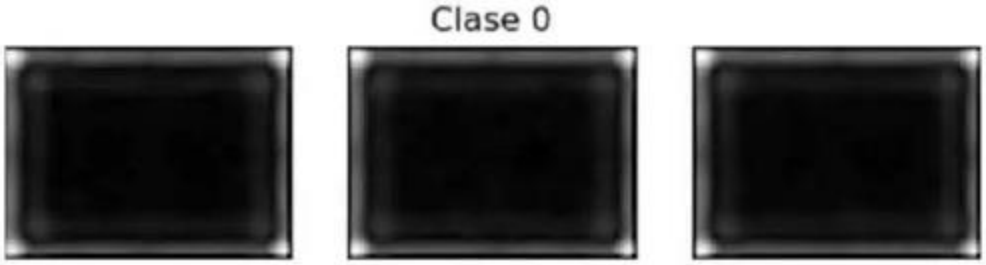


Figura 2. Imagen sin daño por delaminación.

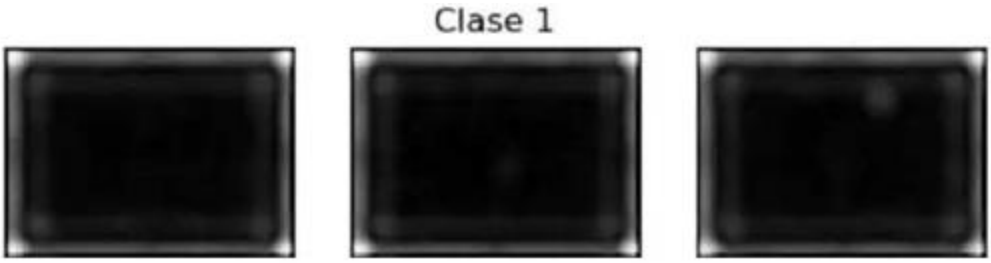


Figura 3. Imagen con daño de delaminación, clase l.

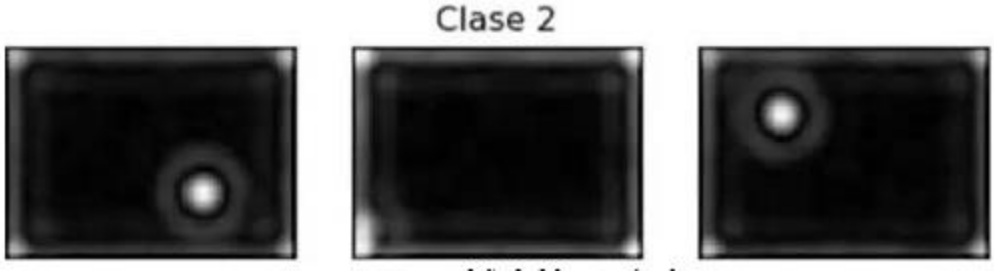


Figura 4. Imagen con daño de delaminación, clase 2.

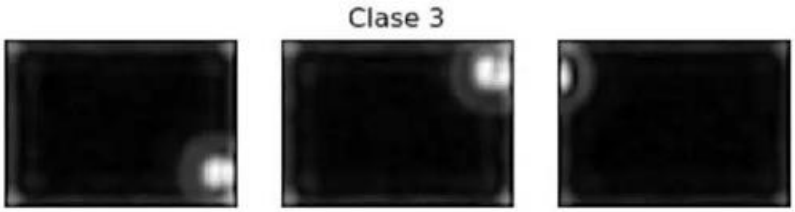


Figura 5. Imagen con daño de delaminación, clase 3.

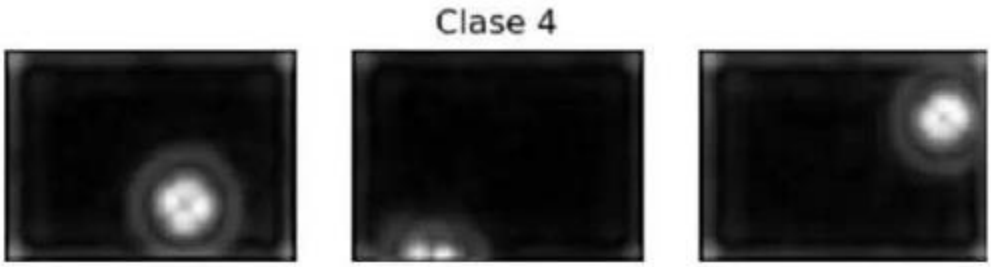


Figura 6. Imagen con daño de delaminación, clase 4.

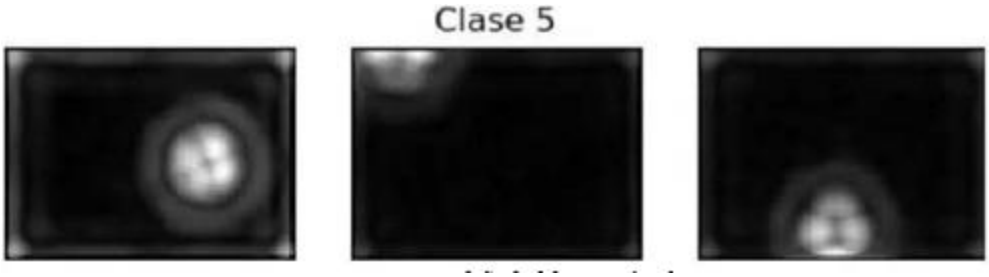


Figura 7. Imagen con daño de delaminación, clase 5.

Para esta tarea se poseen 3500 imágenes diferente con distintos tamaños de daño. El conjunto de entrenamiento posee 2800 imágenes, el de validación posee 175 y el de prueba posee 525. La distribución de imágenes en cada clase para cada conjunto se aprecia en los siguientes histogramas:

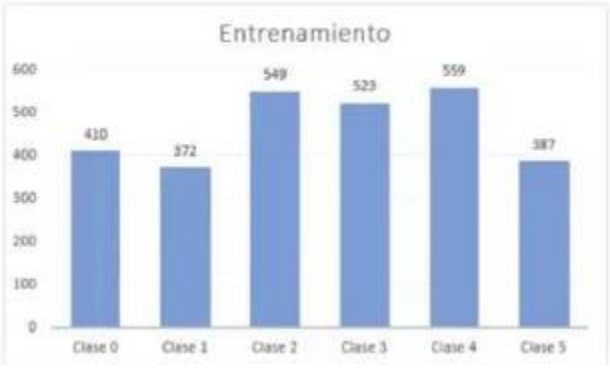


Figura 8. Distribución de datos de entrenamiento.

Figura 9. Distribución de datos de validación.

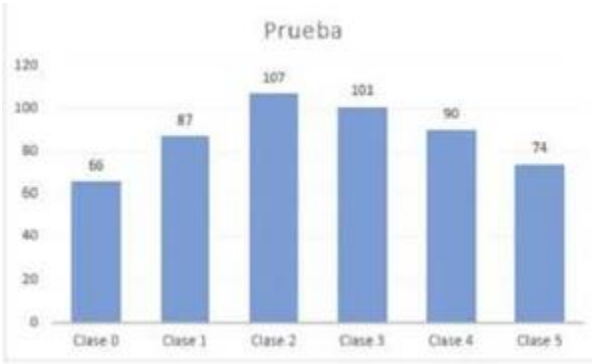


Figura l0. Distribución de datos de prueba.

Para esta aplicación de utilizaron diferentes arquitecturas y modelos para la detección de la delaminación:

1. *One-Class Classification:* Para el entrenamiento de este modelo, debido a que es utilizado para la detección de anomalías solo se utilizan imágenes de clase 0. Para esto los conjuntos de entrenamiento/prueba se distribuyen como 1600/300 imágenes cada uno, mientras que el conjunto de validación consiste en 175 imágenes, donde hay 35 de cada clase desde la 1 a la 5. Esto con el fin de ver la diferencia en el error entre el conjunto de entrenamiento y el de validación a la hora de entrenar.
2. *Clasificación Binaria y Multiclase tipo 2:* Para la distribución de datos de la clasificación binaria, se utilizaron los datos de entrenamiento/validación/prueba de la clasificación One-Class como datos de clase 0 y se combinaron los datos de la clase 1 a la 5 en un nuevo conjunto llamado clase 1.

En este caso solo veremos la *Clasificación Binaria y Multiclase tipo 2* y los resultados obtenidos utilizando este modelo. En primer lugar, se hace una clasificación binaria, se prueba un algoritmo convolucional y uno simple MLP. Posteriormente, se entrena un algoritmo de clasificación multiclase que solo incluye las clases con daño, es decir, clases de la 1 a la 5. A continuación se muestran las arquitecturas usadas en cada caso:

Capa	Tipo de capa	Activación
1	Conv2D(filters = 32, kernel = (7, 7))	ReLu
2	Conv2D(filters = 32, kernel = (7, 7), strides = (2,2), Dropout = 0.3)	ReLu
3	Conv2D(filters = 64, kernel = (5, 5))	ReLu
4	Conv2D(filters = 64, kernel = (5, 5), strides = (2,2), Dropout = 0.3)	ReLu
5	Flatten	-
6	Dense(units = 64, Dropout = 0.3)	ReLu
7	Dense(units = 1)	Sigmoid

Tabla l. Arquitectura de clasificación binaria convolucional

Capa	Tipo de capa	Activación
1	Dense(units = 512, Dropout = 0.3)	ReLu
2	Dense(units = 512, Dropout = 0.3)	ReLu
3	Dense(units = 1)	Sigmoid

Tabla 2. Arquitectura de clasificación binaria MLP.

Capa	Tipo de capa	Activación
1	Conv2D(filters = 32, kernel = (7, 7))	ReLu
2	Conv2D(filters = 32, kernel = (7, 7), strides = (2,2), Dropout = 0.3)	ReLu
3	Conv2D(filters = 64, kernel = (5, 5))	ReLu
4	Conv2D(filters = 64, kernel = (5, 5), strides = (2,2), Dropout = 0.3)	ReLu
5	Flatten	-
6	Dense(units = 64, Dropout = 0.3)	ReLu
7	Dense(units = 5)	Softmax

Tabla 3. Arquitectura para clasificación multiclase tipo 2, CNN.

En las figuras 11 a) y b) se presentan los resultados para la red convolucional de clasificación binaria.

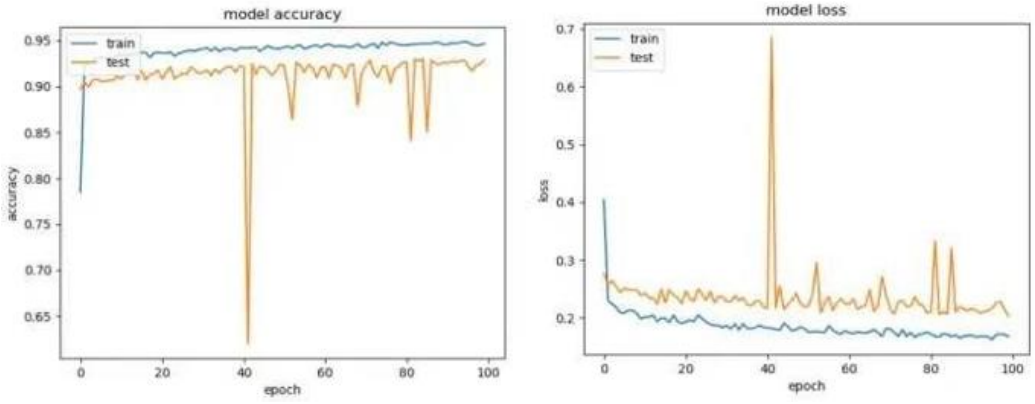


Figura ll. a) Accuracy clasificación binaria convolucional. b) Loss clasificación binaria convolucional.

En la tabla 4 se encuentra el resumen de los resultados de esta red. Se cuenta con el resultado tanto para la fase de entrenamiento como para la fase de prueba.

Entrenamiento				Prueba	
Loss	Accuracy	Val_loss	Val_Accuracy	Loss	Accuracy
0.1676	0.9466	0.2029	0.9285	0.2086	0.9284

Tabla 4. Resultados clasificación binaria convolucional.

Entre las figuras 13 y 14 se presentan los resultados para la red de clasificación binaria del tipo Perceptrón Multicapa (MLP).

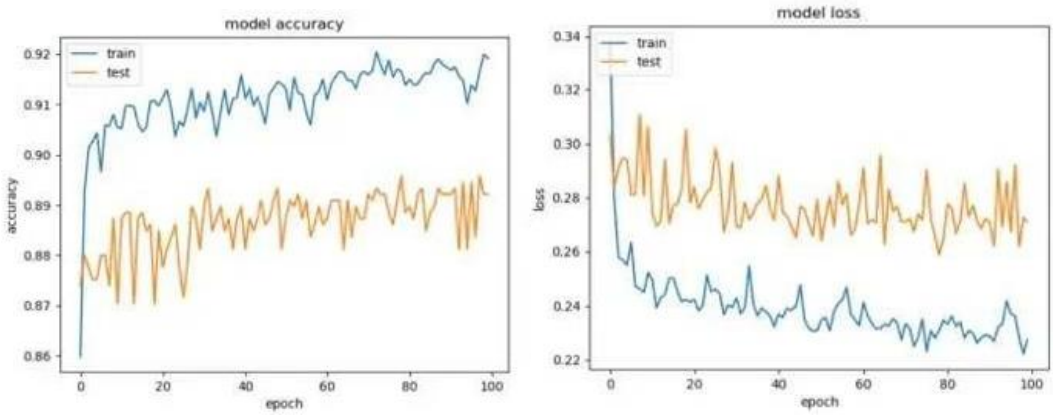


Figura l3. a) Accuracy clasificación binaria MLP. b) Loss clasificación binaria MLP.

En la tabla 5 se encuentra el resumen de los resultados de esta red. Se cuenta con el resultado tanto para la fase de entrenamiento como para la fase de prueba.

Entrenamiento				Prueba	
Loss	Accuracy	Val_loss	Val_Accuracy	Loss	Accuracy
0.2275	0.9191	0.2709	0.8921	0.2708	0.8921

Tabla 5. Resultados clasificación binaria MLP

Se realiza un modelo de clasificación multiclase con el fin de determinar el tamaño de daño, es decir, solo se clasifica entre las clases 1 y 5.

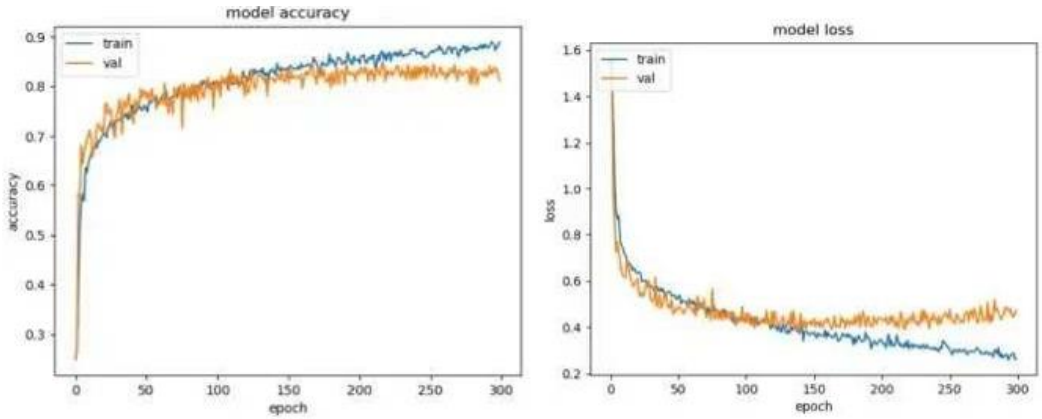


Figura l5. a) Accuracy para clasificación multiclase CNN tipo2. b) Loss para clasificación multiclase CNN tipo 2.

En la tabla 6 se encuentra el resumen de los resultados finales de esta red de clasificación del tamaño de daño, tanto para la fase de entrenamiento como para la fase de prueba.

Fuente: https://repositorio.uchile.cl/bitstream/handle/2250/172881/Detecci%c3%b3n-de-da%c3%b1o-en-una-placa-compuesta-utilizando-variational-autoencoders.pdf?sequence=1&isAllowed=y

https://repositorio.uchile.cl/handle/2250/172881

N° 2 Application of variational autoencoders for aircraft turbomachinery design

En este proyecto, el objetivo es entrenar un codificador automático variacional para modelar las características del flujo de aire supersónico de un álabe del compresor del rotor 37 de la NASA en respuesta a las condiciones cambiantes del flujo másico.

El vector de entrada incluye una imagen de color aplanada que representa los contornos del número de mach relativo así como la condición de contorno de flujo másico asociada. Una vez que se construye el modelo entrenado, novel mach, los diagramas de contorno y las condiciones de contorno se pueden muestrear a partir de la distribución latente posterior. restricciones dadas en la variedad latente. Este enfoque de muestreo de espacio latente proporciona una alternativa a la exploración de espacio de diseño paramétrico donde se optimiza una función de costo utilizando la ingeniería de parámetros directamente y es el racional para seleccionar un VAE.

El conjunto de datos para este proyecto se genera a través de una técnica de simulación numérica conocida como dinámica de fluidos computacional (CFD). Estas simulaciones calculan el campo de flujo alrededor de la forma del perfil aerodinámico en varias condiciones de contorno (un conjunto por simulación). Dos imágenes de entrenamiento de ejemplo se muestran en figura 1. En la imagen de la izquierda, se puede ver una onda de choque cerca del borde de ataque de la superficie de aire, y en

En la imagen de la derecha, la onda de choque es impulsada más profundamente (tragada) en el pasaje de flujo entre las palas. Estos estados constituyen los dos estados principales en el conjunto de datos y los estados principales a ser predichos por el modelo. Junto con la imagen del campo de flujo que muestra los contornos del número de mach relativo, el corregidola condición límite de flujo másico también se guarda para su uso en el entrenamiento

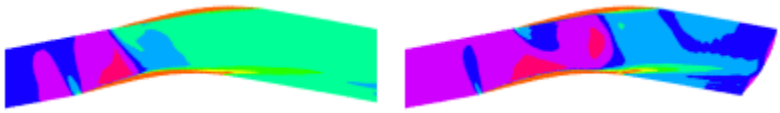


Figura 17: Imágenes de conjuntos de entrenamiento de ejemplo que muestran contornos de máquina relativos. Una onda de choque de vanguardia(izquierda) y se puede ver una onda de choque profunda (tragada) (derecha).

La función objetivo del modelo, que se muestra en la ecuación 1, se basa en maximizar la variación límite inferior que representa el error reconstructivo del codificador automático y la divergencia KL entre la distribución latente estimada $q\varphi(z|x)$ y una distribución previa gaussiana normal, $p\theta(z)$ [1]. La divergencia KLEl término actúa para regularizar el espacio latente y evitar que el modelo simplemente memorice el entrenamiento.conjunto de datos Las redes de codificador y decodificador se utilizan para aproximar las distribuciones posteriores de el espacio latente $q\varphi(z|x)$ y la salida $p\theta(x|z)$.

$$L(\theta, \phi, x^{(i)}) = \frac{1}{L} \sum_{l=1}^{L} (\log p_\theta(x^{(i)}|z^{(i,l)})) - D_{KL}(q_\phi(z|x^{(i)})||p_\theta(z))$$

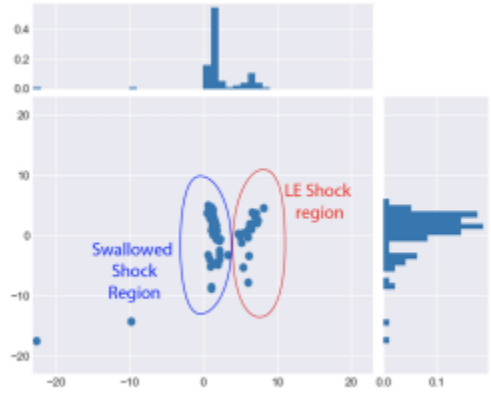


Figura 18: visualización de puntos de prueba codificados en el espacio latente 2D

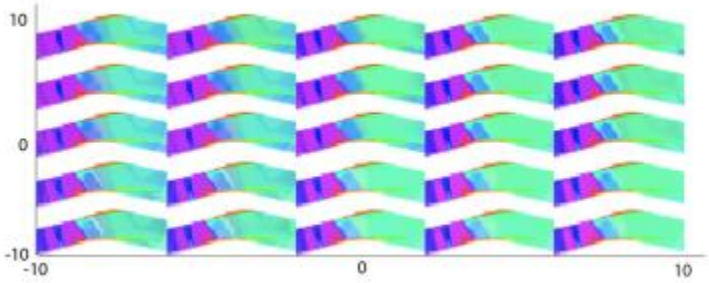


Figura 19: Imágenes reconstruidas como muestra del espacio latente 2D

Una forma popular de medir el rendimiento de los algoritmos VAE es evaluar el registro de probabilidadde la reconstrucción posterior pθ(x|z) [1]. Aunque esto es conveniente para el ajuste de hiperparámetrosy comparando la arquitectura, un enfoque más intuitivo es usar una métrica de similitud entre la entradae imagen reconstruida. Inicialmente, se exploró la métrica del error cuadrático medio (MSE), pero estase determinó que era demasiado sensible para las comparaciones prácticas. Cierta suavidad en el reconstruidose espera el modelo dado el efecto de regularización del término de divergencia KL. En su lugar, se eligió la métrica de similitud estructural de la imagen, que incluye diferencias perceptivas como la textura y la exposición

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

Training Set (4129 samples)	90.5 %
Development Set (498 samples)	92.2 %
Test/Validation Set (245 samples)	90.3 %

Tabla 1: Similitud Reconstructiva (SSIM)

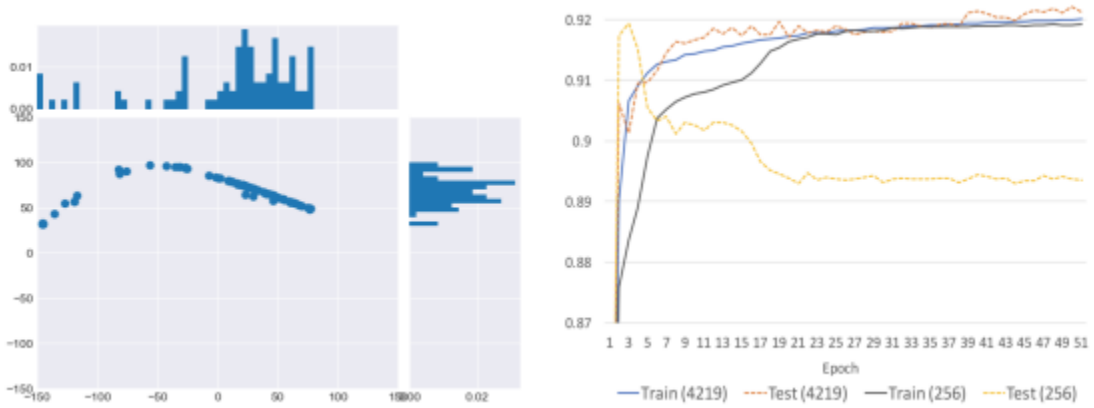


Figura 20: Evidencia de sobreajuste: mala regularización del espacio latente (izquierda) y gran variación entre SSIM de desarrollo de trenes entre tamaños de conjuntos de datos (derecha).

En este proyecto presentamos un enfoque alternativo para la exploración del espacio de diseño de ingeniería. Al entrenar un codificador automático variacional generativo con parámetros funcionales de ingeniería y visualizaciones de campo de flujo, demostramos la capacidad de un modelo para codificar un régimen de flujo transónico complejo y de alta dimensión en un espacio latente bidimensional. Surgieron distintos grupos latentes que representan los regímenes de flujo dominantes en el conjunto de datos. Luego demostramos la capacidad de tomar muestras selectivamente de regiones favorables, creando campos de flujo novedosos con sus condiciones límite de flujo másico asociadas. La métrica SSIM se utilizó como una forma interpretable de medir el rendimiento reconstructivo de los campos de flujo generados y se presentaron resultados coherentes en los conjuntos de datos de entrenamiento, desarrollo y prueba.

Fuente: http://cs229.stanford.edu/proj2017/final-reports/5231979.pdf

2. Realice una investigación de dos posibles aplicaciones donde se usen Redes Adversarias Generativas (GAN) para la generación de algún tipo información. ¿Cuál es la diferencia entre un VAE y una GAN cuando se aplica para la generación de información?

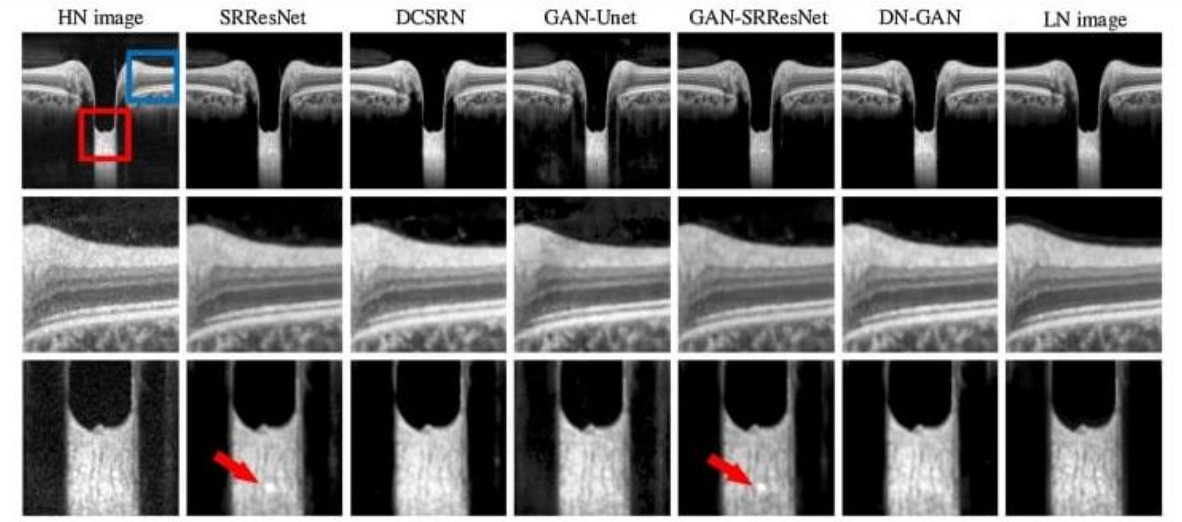
Las aplicaciones de GAN pueden resolver diferentes tareas:

- Generar ejemplos para conjuntos de datos de imágenes
- Traducción de imagen a imagen
- Traducción de texto a imagen
- Traducción semántica de imagen a foto
- Generación de vista frontal de cara
- Genera nuevas poses humanas
- Fotos a Emojis
- Edición de fotografía
- Envejecimiento facial

- Fusión de fotos
- súper resolución
- Repintado de fotos
- Traducción de ropa
- Predicción de vídeo
- Generación de objetos 3D

Ahora es el mejor momento para implementar GAN aprovechando sus habilidades porque pueden modelar distribuciones de datos reales y aprender representaciones útiles para mejorar las canalizaciones de IA, asegurar datos, encontrar anomalías y adaptarse a casos específicos del mundo real.

Eliminación de ruido: eliminación de todo tipo de ruido de los datos. Por ejemplo, eliminar el ruido estadístico de las imágenes de rayos X se ajusta a las necesidades médicas, que se describirán en nuestros casos de uso.



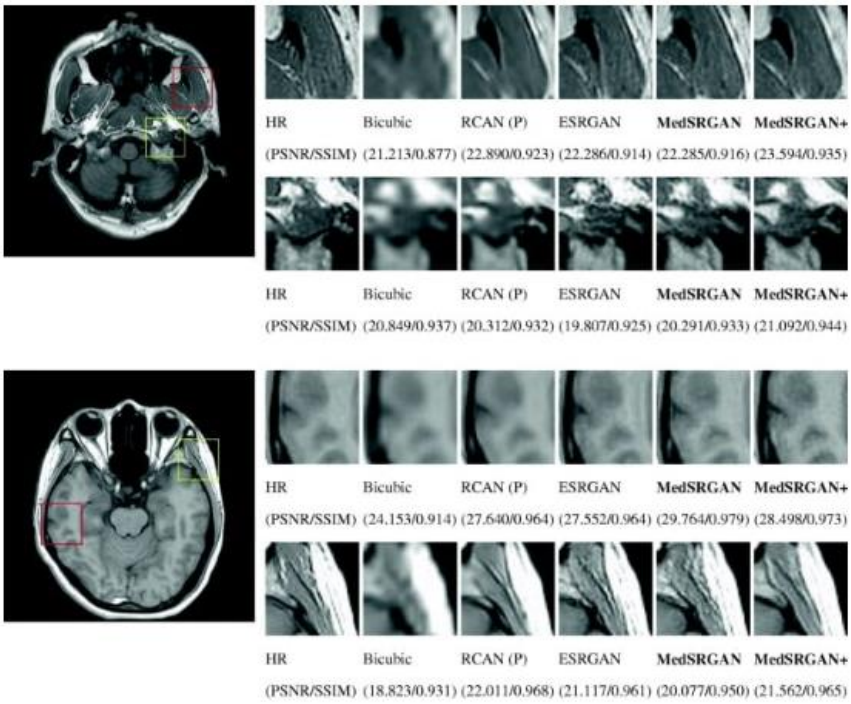
Eliminación de ruido de imágenes de tomografía usando GAN

La posibilidad de mejora de la imagen nos permite implementar GAN en medicina para Super-Resolución fotorrealista de imagen única. ¿Por qué es esto significativo?

El motivo de la gran demanda de GAN en el cuidado de la salud es que las imágenes deben ajustarse a requisitos particulares y ser de alta calidad. La alta calidad de imagen puede ser difícil de obtener bajo ciertos protocolos de medición, por ejemplo, existe una gran necesidad de disminuir el efecto de la radiación en los pacientes cuando se usa el escaneo de dosis baja en la tomografía computarizada (CT, por sus siglas en inglés), para reducir el efecto nocivo en las personas con ciertas precondiciones de salud como el cáncer de pulmón) o resonancia magnética. Tiene el efecto de complicar los esfuerzos para obtener imágenes de buena calidad debido a los escaneos de baja calidad.

La superresolución mejora las imágenes capturadas y puede eliminar bastante bien el ruido; sin embargo, la adopción de GAN en el área médica es bastante lenta, ya que se deben realizar muchos

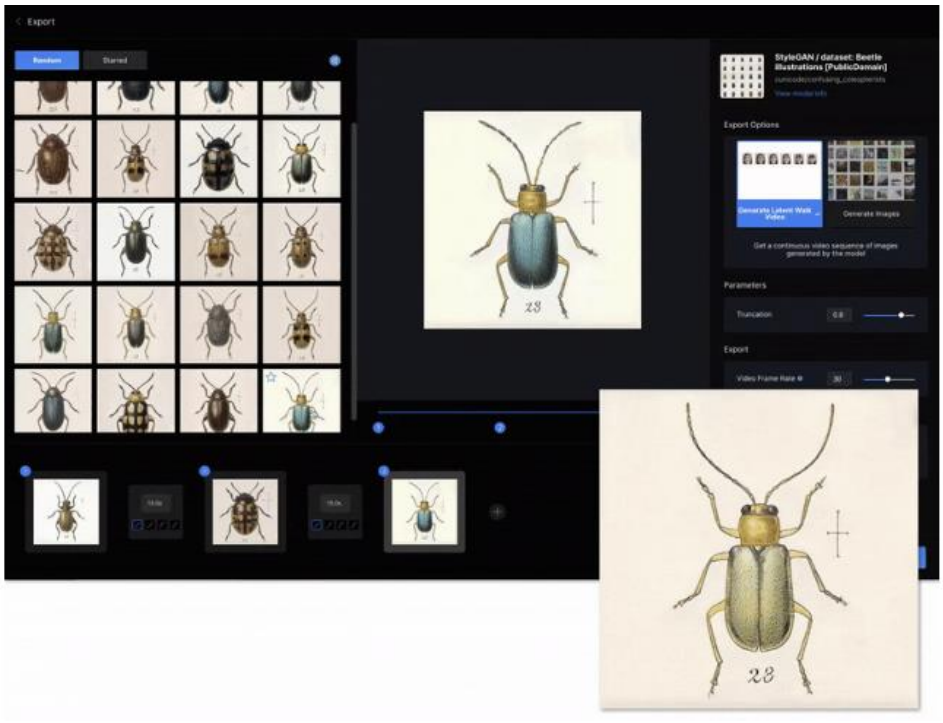
experimentos y pruebas debido a problemas de seguridad. Cuando se trata de atención médica, es obligatorio involucrar a varios expertos en el dominio para evaluar los modelos y garantizar que la eliminación de ruido no distorsione el contenido real de la imagen de alguna manera que pueda conducir a un diagnóstico incorrecto



A pesar de las enormes oportunidades, las GAN tienen problemas. El más grande es su inestabilidad. Las GAN son notoriamente difíciles de entrenar y, a veces, estas redes pueden generar imágenes con artefactos porque los modelos no tienen suficiente información en los datos de entrenamiento para comprender cómo funcionan ciertas cosas en la vida real. Por ejemplo, dado un conjunto de datos de imágenes de retratos, la red puede saber cómo modelar rostros humanos, pero puede fallar en comprender la idea de cómo deben verse los elementos particulares de la ropa. Por lo tanto, es obligatorio elegir cuidadosamente los datos que sean relevantes para el resultado esperado

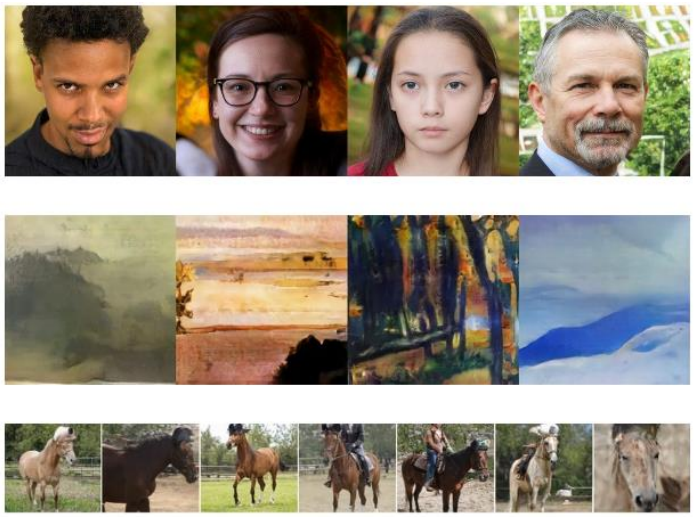
En lugar de encontrar una aplicación de nicho específica para los modelos, algunas empresas ofrecen acceso a GAN y toda la infraestructura e interfaces para manejar los datos, entrenar los modelos y obtener los resultados finales.

Runway AI es una de esas empresas, posicionándose como una plataforma para el aprendizaje automático y permitiendo nuevas técnicas de creación de contenido. Las funciones de medios generativos, como las llama la compañía, son parte de una interfaz web que admite el entrenamiento de un modelo GAN en su propio conjunto de datos y la recopilación de los resultados en forma de imágenes o incluso videos; puede ser muy útil para los creadores de contenido y otros interesados. partes, ya que ayuda a llevar las capacidades de las GAN a las masas (trabajar con GAN sin una interfaz de usuario gráfica puede resultar demasiado inconveniente para la mayoría de los usuarios que no son programadores)



Otras aplicaciones del uso de GAN son:

Generar muestras de datos novedosos, como imágenes de personas, animales, objetos, etc. inexistentes. De esta manera, no solo se pueden generar imágenes, sino también otros tipos de medios (audio, texto)

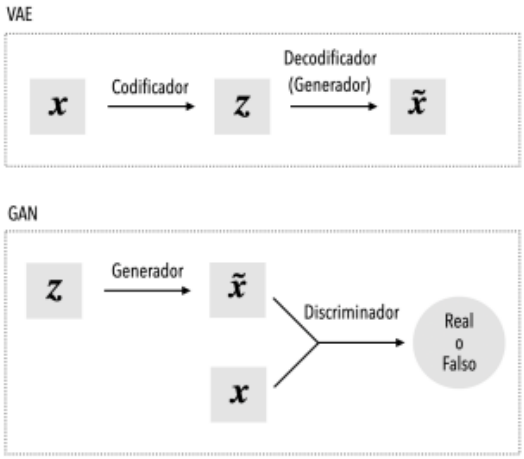


Repintado de imágenes: restauración de partes faltantes de imágenes

Superresolución de imagen : escalado de imágenes de baja resolución a alta resolución sin artefactos de escalado notables.

Fuente: https://mobidev.biz/blog/gan-technology-use-cases-for-business-application

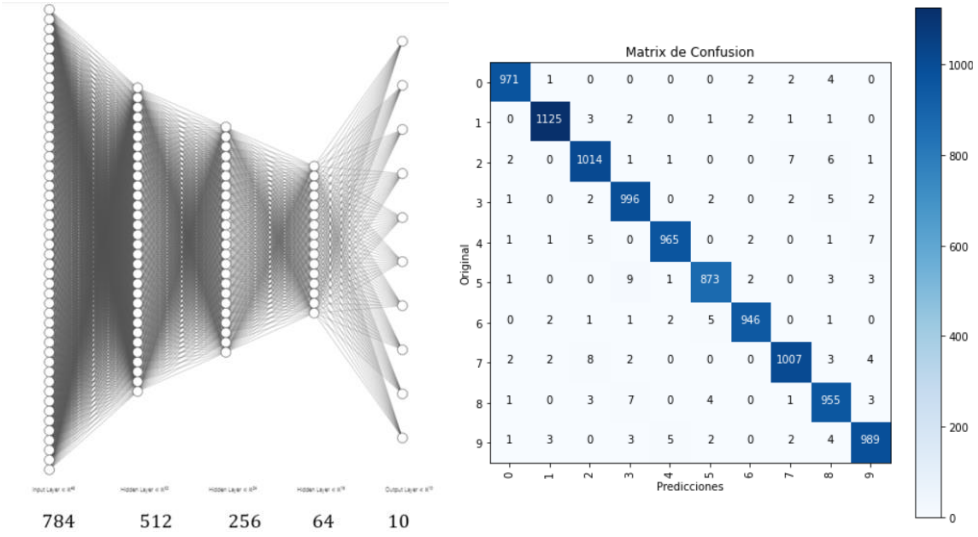
Las diferencias fundamentales entre los VAEs y las GANs son que para los primeros existe una única función de pérdida mientras que para las redes GANs se tienen dos funciones de pérdida, una para el generador y otra para el discriminador. Por otro lado, para los VAEs se tiene una función de pérdida que se pretende optimizar (minimizar), pero en el caso de las redes GANs, no se cuenta con una métrica explícita que se pretenda optimizar. En su lugar, se tienen dos objetivos que compiten entre ellos y que no se pueden traducir en una única función a optimizar



Fuente: http://sedici.unlp.edu.ar/bitstream/handle/10915/101507/Documento_completo.pdf-PDFA.pdf?sequence=1&isAllowed=y#:~:text=Las%20diferencias%20fundamentales%20entre%20los,y%20otra%20para%20el%20discriminador.

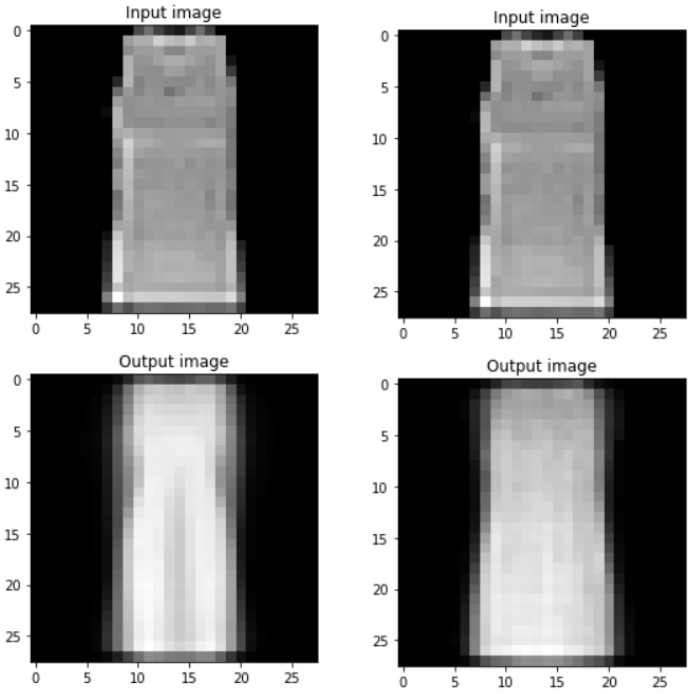
3. Realice una aplicación de clasificación utilizando Autoencoders apilados usando la base de datos MNIST

Para la aplicación de clasificación utilizando Autoencoders apilados, se realiza inicialmente el entrenamiento de 3 Autoencoders de forma independiente considerando que las dimensiones de entradas entre una capa latente y la otra sean equivalentes entre si, se tiene uno simple o básico de 784 – 512 – 784, otro de 784 – 512 – 256 – 512 – 784 y el ultimo de 784 – 256 – 64 – 256 – 784, de los cuales se extrae las respectivas capas latentes para pasar por una última capa densa de 10 neuronas con función de activación softmax. Proceso que concluye con una matriz de confusión, en la cual se observa resultados realmente buenos frente a la cantidad de elementos clasificados correctamente. En el siguiente enlace se presenta el Colab.



4. Entrene un Autoencoder para visualizar en dos dimensiones el comportamiento de la base de datos Fashion MNIST. Realice una visualización similar usando PCA (Análisis de Componentes Principales) y compare los resultados obtenidos. Adicionalmente, verifique la capacidad de generar nuevas imágenes con el Autoencoder entrenado.

Para la visualización en dos dimensiones se realiza el entrenamiento de un autoencoder con una capa latente de 2, estructurado de la siguiente forma 784 – 512 – 256 – 128 – 64 – 16 – 2 – 16 – 64 – 128 – 256 – 512 – 784 y se realiza el entrenamiento con las mismas imágenes para la entrada y el target. Posteriormente, se realiza la visualización de la reconstrucción realizada por el autoencoder. Por último, con la reducción de dimensionalidad generada por el PCA se realiza el entrenamiento del decoder con el PCA de entrada y las imágenes como target, del cual se obtiene la predicción de la misma imagen con el fin de comparar los resultados. Al comparar los resultados presentados en las imágenes se puede observar que la respuesta del PCA presenta una mayor distorsión a la obtenida en la disminución de dimensionalidad por medio del Autoencoder. En el siguiente enlace se presenta el Colab.



5. Realice una aplicación de clasificación utilizando un Autoencoders apilados con la base de datos Fashion MNIST

Enlace a la base de datos Fashion MNIST
https://github.com/zalandoresearch/fashion-mnist/tree/master/data/fashion

Para la aplicación de clasificación utilizando Autoencoders apilados, se toma como base la estucura mencionada en el punto 3, con 3 Autoencoders, estructurados con uno básico de 784 – 512 – 784, otro de 784 – 512 – 256 – 512 – 784 y el ultimo de 784 – 256 – 64 – 256 – 784, de los cuales se extrae las respectivas capas latentes para pasar por una última capa densa de 10 neuronas con función de activación softmax. Proceso que concluye con una matriz de confusión, en la cual se observa resultados realmente buenos frente a la cantidad de elementos clasificados correctamente. En el siguiente enlace se presenta el Colab.