

# Informe Final

Tema 2 - Estimación de mapas de oclusión en redes neuronales

Integrantes: André Teixeira  
Raydel Ortigueira  
Profesor: Pablo Estévez V.  
Auxiliar: Ignacio Reyes J.  
Ayudantes: Esteban Reyes D.  
Germán García Jara  
Mauricio Jofre  
Miguel Videla A.  
Nicolás Tapia Rivas

Fecha de realización: 9 de diciembre de 2019

Fecha de entrega: 9 de diciembre de 2019

Santiago, Chile

# Índice de Contenidos

<b>1. Introducción</b>	<b>1</b>
<b>2. Marco Teórico</b>	<b>3</b>
2.1. Conceptos, planteamiento del problema y resultado deseado. . . . .	3
2.2. Criterio de detención . . . . .	3
2.3. Lista de Parámetros . . . . .	4
2.4. Preprocesamiento de los datos . . . . .	4
2.5. Autoencoder . . . . .	5
<b>3. Metodología Experimental</b>	<b>6</b>
3.1. Base de datos . . . . .	6
3.2. Descripción de experimentos . . . . .	8
3.3. Métricas . . . . .	9
3.4. Recursos computacionales . . . . .	10
<b>4. Resultados y discusión</b>	<b>11</b>
4.1. Experimentos . . . . .	11
4.2. Mejor modelo . . . . .	17
<b>5. Conclusiones</b>	<b>18</b>
<b>Referencias</b>	<b>19</b>

# Índice de Figuras

1. Ejemplo de mapa de oclusión . . . . .	1
2. Desplazamiento de los parches que ocultan información de la imagen . . . . .	1
3. Autoencoder convolucional genérico para solucionar el problema . . . . .	2
4. Arquitectura del primer autoencoder utilizado sin condicionamiento . . . . .	5
5. Muestra de 3 ejemplos de las imágenes que contiene la base de datos . . . . .	7
6. Ejemplo Positivo de una Imagen de Supernova . . . . .	7
7. Ejemplo Negativo de una Imagen de Supernova . . . . .	7
8. Sumario del numero de parametros en cada layer y su total . . . . .	10
9. Ejemplos seleccionados correspondientes a modelo entrenado con No Supernovas . . .	12
10. Ejemplos seleccionados correspondientes a modelo entrenado con Supernovas . . . .	13
11. Resultados de MSE de validación y entrenamiento promedio para diferentes tipos de autoencoders . . . . .	14
12. Resultados de Loss por epoca en entrenamiento para el mejor modelo del autoencoder	15
13. Resultados de Loss por epoca de validación para el mejor modelo del autoencoder . .	15
14. Arquitectura del mejor modelo de Autoencoder, con condicionamiento . . . . .	17

## Índice de Tablas

1.	Lista de Parámetros, Hiper-Parámetros y Argumentos . . . . .	4
2.	Dimensiones de los datos . . . . .	6
3.	Estadísticas de los datos . . . . .	6
4.	Valores de MSE para modelo de autoencoder normal y autoencoder con pesos congelados en el encoder . . . . .	11
5.	Modelos ordenados de mayor a menor según MSE de validación . . . . .	11
6.	Valores de MSE para modelo concatenado con etiquetas Numeral y etiquetas One-hot encoded . . . . .	14
7.	Muestras para modelo congelado . . . . .	16
8.	Muestras para modelo condicionado . . . . .	16
9.	Estadísticas para comparar modelo de pesos congelados no condicionado y pesos congelados condicionado . . . . .	16

# 1. Introducción

Una de las técnicas que han surgido para visualizar el comportamiento interno de modelos de clasificación es el mapa de oclusión. El mapa de oclusión de una imagen arbitraria es un mapa que indica cuales regiones de la imagen son más importantes para el modelo cuando efectúa la clasificación (Figure.1). Este mapa se genera al calcular la certeza que produce el clasificador para la clase de la imagen analizada, cuando a la entrada de este se colocan varias versiones de la imagen, donde cada una de ellas tiene tapada una región distinta, la cual se va desplazando a través de la imagen (Figure.2).

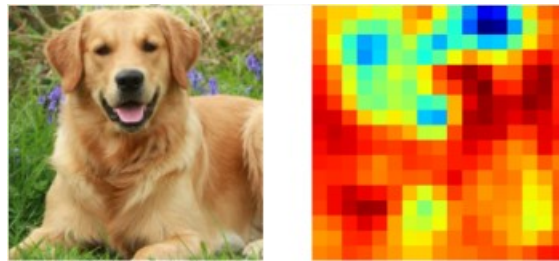


Figura 1: Ejemplo de mapa de oclusión

La omisión de información de la imagen para el clasificador (Figure.2) causará un impacto en el valor de la certeza al clasificar la imagen, por lo que si son omitidas las regiones de la imagen que más información útil aportan al clasificador el valor de la certeza del modelo será pobre. El resultado de estas predicciones iterativas, causadas por el desplazamiento del parche que oculta información, será un mapa que mostrará las regiones de la imagen donde las certezas más bajas, correspondiendo a las regiones más importantes que necesita el modelo para clasificar correctamente.

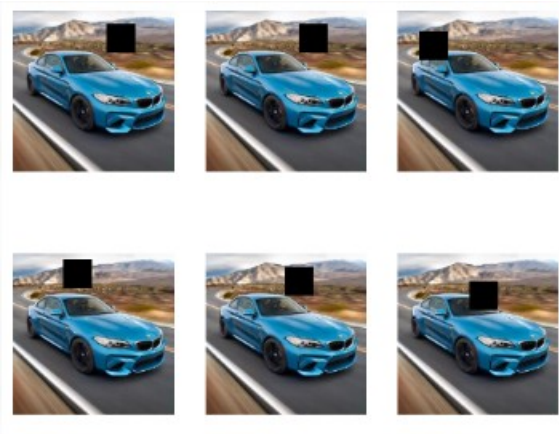


Figura 2: Desplazamiento de los parches que ocultan información de la imagen

Cada pixel del mapa de oclusión es obtenido cuando el modelo efectúa una inferencia por cada posición del parche que oculta información de la imagen. Por lo tanto, si la imagen cuenta con  $N \times N$  pixeles, entonces el mapa requiere de  $N \times N$  inferencias del modelo, por lo que el problema de calcular un mapa de oclusión es que consume muchísimo tiempo.

Mediante este proyecto se buscará estimar mapas de oclusión a través de un autoencoder, el cual luego de ser entrenado reduciría el número de inferencias necesarias de  $N \times N$  a 1.

Por lo tanto, el objetivo principal de este trabajo es obtener un autoencoder convolucional con el cual sea posible estimar mapas de oclusión y de esta forma reducir el tiempo de obtención de los mismos (Figure. 3).

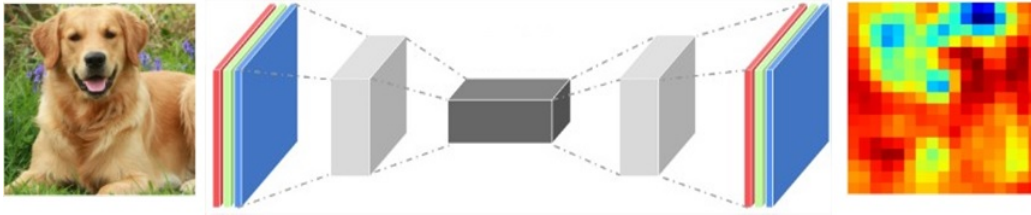


Figura 3: Autoencoder convolucional genérico para solucionar el problema

## 2. Marco Teórico

### 2.1. Conceptos, planteamiento del problema y resultado deseado.

#### Autoencoder

Un autoencoder es una red neuronal artificial no supervisada que aprende a comprimir y codificar eficientemente los datos, luego aprende a reconstruir los datos de la representación codificada reducida a una representación lo más cercana posible a la entrada original. El entrenamiento luego implica el uso de la propagación hacia atrás para minimizar la pérdida de reconstrucción de la red.

#### Problema

Elevada demora en el cálculo de mapas de oclusión debido al tiempo de procesamiento que requiere un modelo para realizar una inferencia por cada posición del parche.

#### Resultado deseado

Si un autoencoder es entrenado para estimar mapas de oclusión y aprende a estimar correctamente, entonces podrá ser utilizado para calcular estos mapas y reducir el tiempo de procesamiento.

### 2.2. Criterio de detención

Como criterios de detención fueron utilizados dos:

- **Número de Epochs:** es el número de veces que el programa se repite, donde en cada época tenemos 30000 iteraciones que corresponden al número de muestras de entrenamiento.
- **EarlyStopping:** este criterio finaliza el entrenamiento para evitar el sobreajuste y que los resultados empeoren, donde el criterio de parada se estableció cuando la curva de validación de la loss function alcanzara su valor mínimo.

EarlyStopping es usado para evitar el sobreajuste y por ende que los resultados empeoren.

**EarlyStopping(`monitor='val_loss', mode='min', patience=3, verbose=1`)**

Aquí finaliza el entrenamiento si encuentra un mínimo en los valores de loss en la validación. Con el parámetro **paciente** igualado a 3 se evita la detención en un mínimo local durante el inicio del entrenamiento y también el progreso.

## 2.3. Lista de Parámetros

Tabla 1: Lista de Parámetros, Hiper-Parámetros y Argumentos

Parámetro	Valor Por Defecto	Descripción
tipo	3	Tipo de Clase de Imagen
encoding_dim	36	Número de características en la densa del Bottleneck
filter_size	3	Tamaño del filtro cuando se hace la convolución
pool_size	2	Cuanto se quiere hacer de Pooling
batch	10	Tamaño de los minibatches
validation_split	0.25	Porcentage de la base datos para hacer validación
model		Cargar el modelo del clasificador para congelar los pesos
input_images		imágenes de entrada para entrenamiento
labels_t		Labels para condicionamiento en forma Numeral
labels_t_cat		Labels para condicionamiento en forma One-Hot Encoding

## 2.4. Preprocesamiento de los datos

Como presupuesto para este proyecto fue necesario contar con mapas de oclusión correspondientes a las imágenes seleccionadas de la base de datos. Por lo tanto, hubo que elaborar una base de datos de mapas de oclusión, para la cual fue conformado y entrenado un clasificador [5].

Para evitar la aparición de valores pequeños en los bordes de los mapas de oclusión calculados y mantener el tamaño de las imágenes originales fue utilizada la técnica de padding, mediante la cual fueron rellenados con ceros los bordes de las imágenes originales. Así, al pasar del cuello de botella a la salida, se tiene una imagen de 24x24 que se puede recortar.

## 2.5. Autoencoder

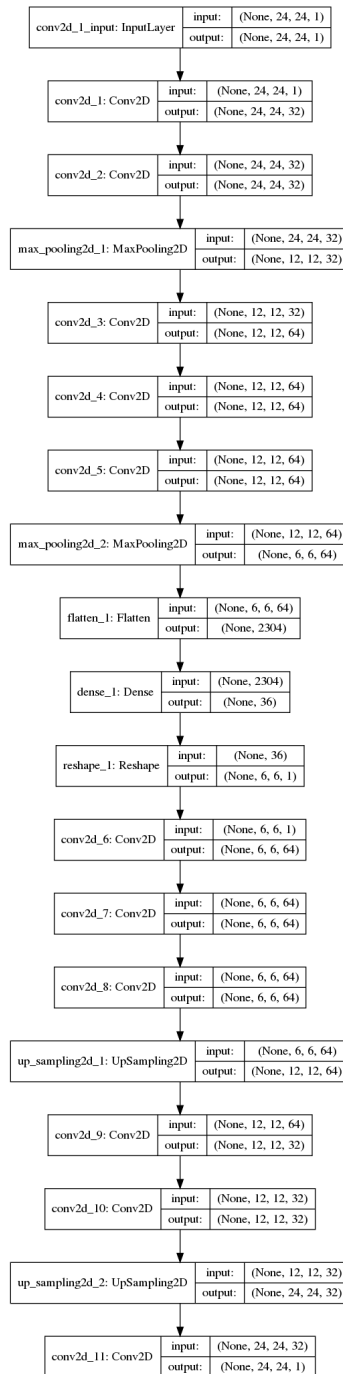


Figura 4: Arquitectura del primer autoencoder utilizado sin condicionamiento

Las capas de este autoencoder son las mismas capas convolucionales que las del clasificador espejadas, pero con un "bottleneck", una capa densa de 36 características.



### 3. Metodología Experimental

#### 3.1. Base de datos

Los datos utilizados en el proyecto corresponden a imágenes en escalas de grises pertenecientes a observaciones astronómicas, clasificadas en 2 categorías: Supernovas y No Supernovas. La base de datos [3] cuenta con 100.000 ejemplos y cada ejemplo cuenta con 4 tipo de imágenes (Figura 5) llamadas Template, Science, Difference y SNR Difference, sumando un total de 400.000 imágenes en la base de datos.

Fueron extraídas de la base de datos sus dimensiones y cuantificados algunos aspectos de interés, además fue comprobado que la base de datos se encuentra balanceada (Tabla 2). También fueron extraídas algunas estadísticas de interés donde se evidencia que los datos están normalizados entre 0 y 1 (Tabla 3).

Tabla 2: Dimensiones de los datos

Dimensiones de los datos	Valores
Dimensiones de la base de datos	100000x21x21x4
Dimensiones de las imágenes	21x21x1
Número de ejemplos	100000
Cantidad de tipos de imágenes	4
Cantidad de ejemplos positivos	50000
Cantidad de ejemplos negativos	50000

Tabla 3: Estadísticas de los datos

Estadísticas de los datos	Valores
Valor máximo	1
Valor mínimo	0
Valor promedio	0.436
Valor de varianza	0.044
Valor de desviación estándar	0.211

Para el proyecto solo fueron seleccionadas las imágenes del tipo 'SNR Difference', las cuales representan el 25 % de la base de datos. Para confeccionar los mapas de oclusión fue entrenado un clasificador tomando el total de las imágenes seleccionadas.

Para el entrenamiento de los modelos que fueron utilizados en los experimentos fueron creadas 3 bases de datos extras [7], a partir de las imágenes seleccionadas. La primera está conformada por las primeras 20.000 imágenes etiquetadas como Supernova, la segunda está conformada por las primeras 20.000 imágenes etiquetadas como No Supernova y la tercera está conformada por la unión de las dos bases de datos anteriores sumando un total de 40.000 imágenes mixtas.

Cada una de las 3 bases de datos anteriores contiene 2 tipos de imágenes: las imágenes originales y sus correspondientes mapas de oclusión.

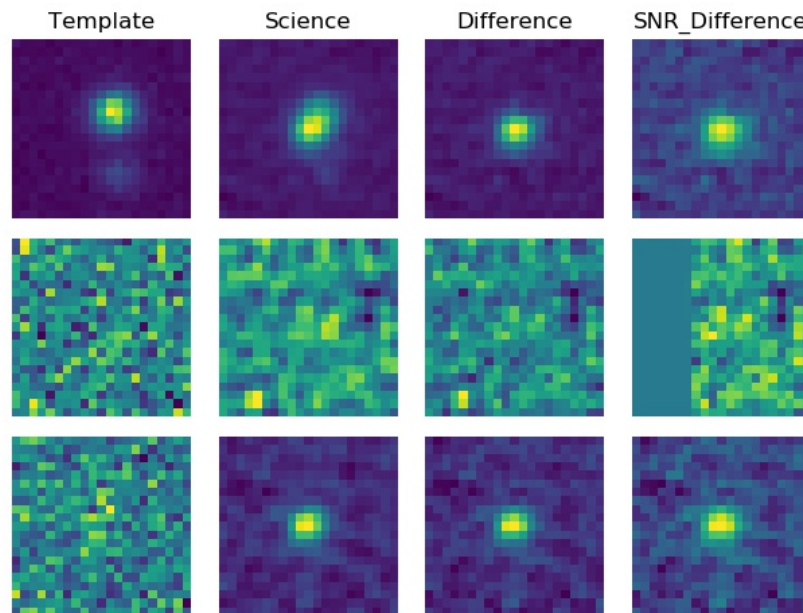


Figura 5: Muestra de 3 ejemplos de las imágenes que contiene la base de datos

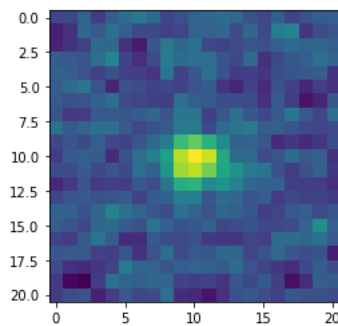


Figura 6: Ejemplo Positivo de una Imagen de Supernova

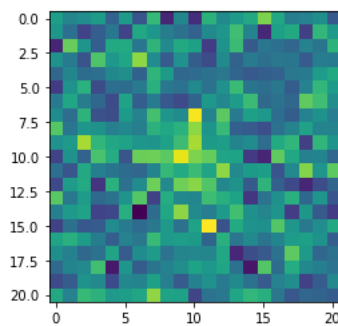


Figura 7: Ejemplo Negativo de una Imagen de Supernova

### 3.2. Descripción de experimentos

Como presupuesto para los experimentos fueron conformados 2 autoencoders convolucionales, uno sin entrenamiento previo que será llamado Normal y otro que cuenta en etapa del encoder con los pesos pertenecientes al modelo clasificador utilizado en la elaboración de los mapas de ocusión, para este último modelo solo serán entrenados los pesos pertenecientes a la etapa del decoder quedando sin entrenar los pesos de la etapa del encoder, este modelo será llamado Congelado. A continuación serán explicados cada uno de los experimentos llevados a cabo.

La base de datos fue dividida en 2 partes, un conjunto de entrenamiento conformado por el 75 % de los ejemplos disponibles y un conjunto de validación conformado por el 25 % restante. La librería utilizada fue Keras y el código escrito por terceros se encuentra en [1,6].

#### Experimento No.1

Este experimento consiste en entrenar cada uno de los 2 autoencoders disponibles con cada una de las 3 bases de datos creadas durante el proyecto, por lo que se efectuarán 6 entrenamientos. Para cada uno de estos serán calculados los MSE promedio de entrenamiento y de validación. Luego, basados en el MSE de validación serán comparados los entrenamientos para determinar qué pareja de modelo entrenado y base de datos obtiene el menor valor de MSE.

Luego deben ser seleccionados el mejor modelo obtenido con la base de datos Supernova y el mejor modelo obtenido con la base de datos No Supernova. Además, para cada modelo seleccionado anteriormente deben ser mostrados los 2 ejemplos con mayor valor de MSE y los 2 ejemplos con menor valor de MSE.

#### Experimento No.2

Este experimento consiste en agregar una característica al cuello de botella del autoencoder, la cual corresponde al valor de la etiqueta perteneciente a la imagen que se esté procesando. La característica agregada tendrá 2 formatos con los cuales se conformarán 2 modelos: uno que será el valor exacto de la etiqueta y otro que será la codificación One-Hot de dicha etiqueta. Ambos modelos serán entrenados y además serán calculados sus valores de MSE de entrenamiento y de validación, luego será seleccionado el modelo de mejores resultados.

#### Experimento No.3

Este experimento consiste en entrenar varios autoencoders resultantes de seccionar el autoencoder original simétricamente, de forma tal que la porción de autoencoder a la izquierda del cuello de botella sea la misma que la porción existente a la derecha del cuello.

Los cortes realizados en el autoencoder original fueron ejecutados en las posiciones correspondientes a las capas de Max Pooling. El primer corte fue hecho en las últimas 3 capas convolucionales del clasificador implementado previamente, o sea, tiene las capas convolucionales hasta el segundo Maxpooling. El segundo corte se hizo en la segunda capa convolucional, antes del primero Maxpooling del clasificador, quedando solo 1 capa del clasificador.

Fueron realizados 2 cortes y entrenados 3 modelos incluyendo el autoencoder original. Luego de ser entrenados los 3 modelos, es seleccionado el modelo de menor valor de MSE de validación y además son mostradas sus curvas de costo de validación y de entrenamiento.

#### Experimento No.4

Este experimento consiste en seleccionar el mejor modelo de cada uno de los experimentos ante-

riores y entrenarlos 5 veces cada uno. Para cada entrenamiento se deben obtener los MSE promedio de entrenamiento y de validación. Luego, se debe calcular el valor promedio y la desviación estándar para cada modelo basado en los resultados obtenidos en sus 5 entrenamientos, y basado en esas estadísticas se debe seleccionar el mejor modelo.

### 3.3. Métricas

Medidas de desempeño para evaluar los modelos:

- MSE promedio y su variación estandar
- Valores de la función de loss

Con el fin de identificar el mejor modelo para este proyecto, fue seleccionado el que obtuvo el valor más bajo de error cuadrático medio, es decir, el promedio de los cuadrados de las diferencias entre los valores de los píxeles del mapa de oclusión predicho y el mapa de oclusión original. Para esto, fue utilizado el valor medio de los resultados de MSE entre varias sesiones de entrenamiento. Sin embargo, solo el valor promedio no es suficiente para la toma de decisiones, por lo que fue considerada también la desviación estándar de estos resultados. Además, se requiere un menor valor de loss tanto en entrenamiento como en validación para verificar la efectividad del modelo.

### 3.4. Recursos computacionales

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 24, 24, 32)	544
conv2d_2 (Conv2D)	(None, 24, 24, 32)	9248
max_pooling2d_1 (MaxPooling2D)	(None, 12, 12, 32)	0
conv2d_3 (Conv2D)	(None, 12, 12, 64)	18496
conv2d_4 (Conv2D)	(None, 12, 12, 64)	36928
conv2d_5 (Conv2D)	(None, 12, 12, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(None, 6, 6, 64)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 36)	82980
reshape_1 (Reshape)	(None, 6, 6, 1)	0
conv2d_6 (Conv2D)	(None, 6, 6, 64)	640
conv2d_7 (Conv2D)	(None, 6, 6, 64)	36928
conv2d_8 (Conv2D)	(None, 6, 6, 64)	36928
up_sampling2d_1 (UpSampling2D)	(None, 12, 12, 64)	0
conv2d_9 (Conv2D)	(None, 12, 12, 32)	18464
conv2d_10 (Conv2D)	(None, 12, 12, 32)	16416
up_sampling2d_2 (UpSampling2D)	(None, 24, 24, 32)	0
conv2d_11 (Conv2D)	(None, 24, 24, 1)	289
Total params: 294,789		
Trainable params: 192,645		
Non-trainable params: 102,144		

Figura 8: Sumario del numero de parametros en cada layer y su total

En la Figura 8 se muestra el número de pesos o parámetros en cada capa del autoencoder. Fue verificado que los parámetros entrenables totales suman 192,645 pesos y los parámetros no entrenables 102,144 pesos, siendo estos últimos los pesos de las capas convolucionales congeladas copiados de los pesos de las capas convolucionales del clasificador. Obteniendo un total de 294,789 pesos.

Estableciendo que 1 peso ocupa 4 bytes, entonces el espacio en memoria que ocupará el modelo será de 1179156 bytes.

## 4. Resultados y discusión

### 4.1. Experimentos

El código perteneciente a los experimentos y que replica los resultados se encuentra en [5].

#### Experimento No.1

La Tabla 4 muestra los valores de MSE de entrenamiento y de validación obtenidos del experimento y la Tabla 5 muestra los valores anteriores ordenados de mayor a menor tomando en cuenta el MSE de validación, donde se refleja que el modelo congelado entrenado con Supernova obtuvo el menor valor de MSE de validación con un valor de 0.01170, por lo que fue seleccionado como el mejor de las 6 propuestas.

Con respecto a los modelos entrenados con Supernovas se tiene que el mejor modelo coincide con el mejor modelo de todos los entrenados y seleccionado anteriormente. Por otra parte, con respecto a los modelos entrenados con No Supernovas se tiene que el modelo congelado fue el que obtuvo el menor valor MSE de validación con un valor de 0.01525, por lo que también fue seleccionado.

Los modelos anteriores, con los cuales se obtuvieron los valores de MSE más bajos para sus respectivas bases de datos, coinciden en poseer el modelo congelado, el cual contiene en su etapa de encoder los pesos heredados del clasificador utilizado en la elaboración de los mapas de oclusión. Estos pesos hicieron que el clasificador obtuviera un alto rendimiento, por lo que al ser transferidos al autoencoder y no ser entrenados favorecieron el rendimiento de este nuevo modelo.

Tabla 4: Valores de MSE para modelo de autoencoder normal y autoencoder con pesos congelados en el encoder

Modelo	BBDD	MSE train	MSE val
Normal	Datos mixtos	0.02284	0.02288
	Super Nova	0.01398	0.01369
	No Super Nova	0.01921	0.01981
Congelado	Datos mixtos	0.01452	0.01525
	Super Nova	0.01155	0.01170
	No Super Nova	0.01389	0.01525

Tabla 5: Modelos ordenados de mayor a menor según MSE de validación

Modelo	BBDD	MSE train	MSE val
Normal	Datos mixtos	0.02284	0.02288
Normal	No Super Nova	0.01921	0.01981
Congelado	Datos mixtos	0.01452	0.01525
<b>Congelado</b>	<b>No Super Nova</b>	<b>0.01389</b>	<b>0.01525</b>
Normal	Super Nova	0.01398	0.01369
<b>Congelado</b>	<b>Super Nova</b>	<b>0.01155</b>	<b>0.01170</b>

Luego, fueron extraídos los 2 ejemplos con los valores de MSE de validación más bajos y los 2 ejemplos con los valores de MSE de validación más altos de cada uno de los modelos seleccionados.

En las Figuras 9 y 10 son mostrados los ejemplos extraídos, donde para ambas figuras las 2 imágenes ubicadas en la parte superior corresponden a los ejemplos con mayor error y las ubicadas en la parte inferior corresponden a los ejemplos con menor error.

Los altos valores de MSE son generalmente producto de predicciones donde los valores de probabilidad de sus píxeles son casi opuestos en su totalidad a la etiqueta correspondiente (Figura 9 (a) y Figura 10 (a,b)) y de predicciones correspondientes a otras etiquetas (Figura 9(b,d)).

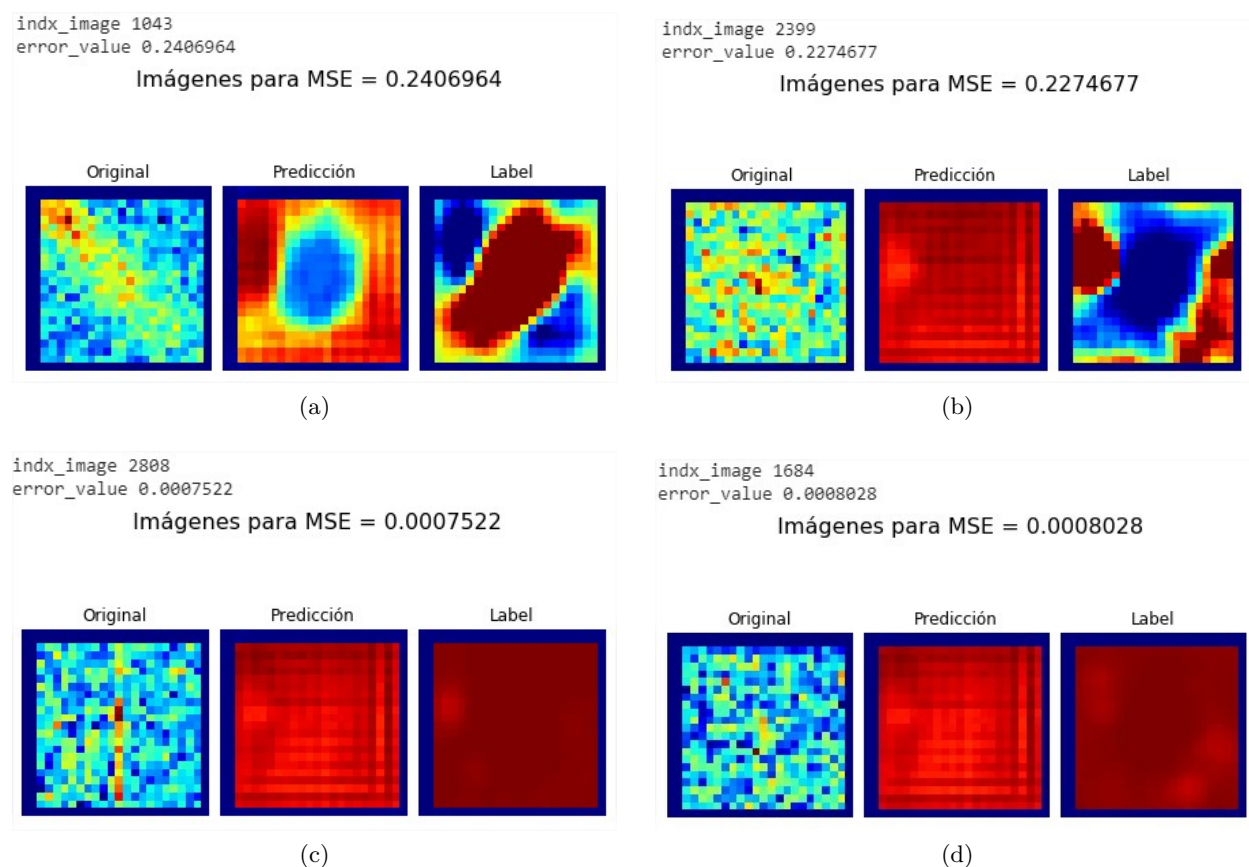


Figura 9: Ejemplos seleccionados correspondientes a modelo entrenado con No Supernovas

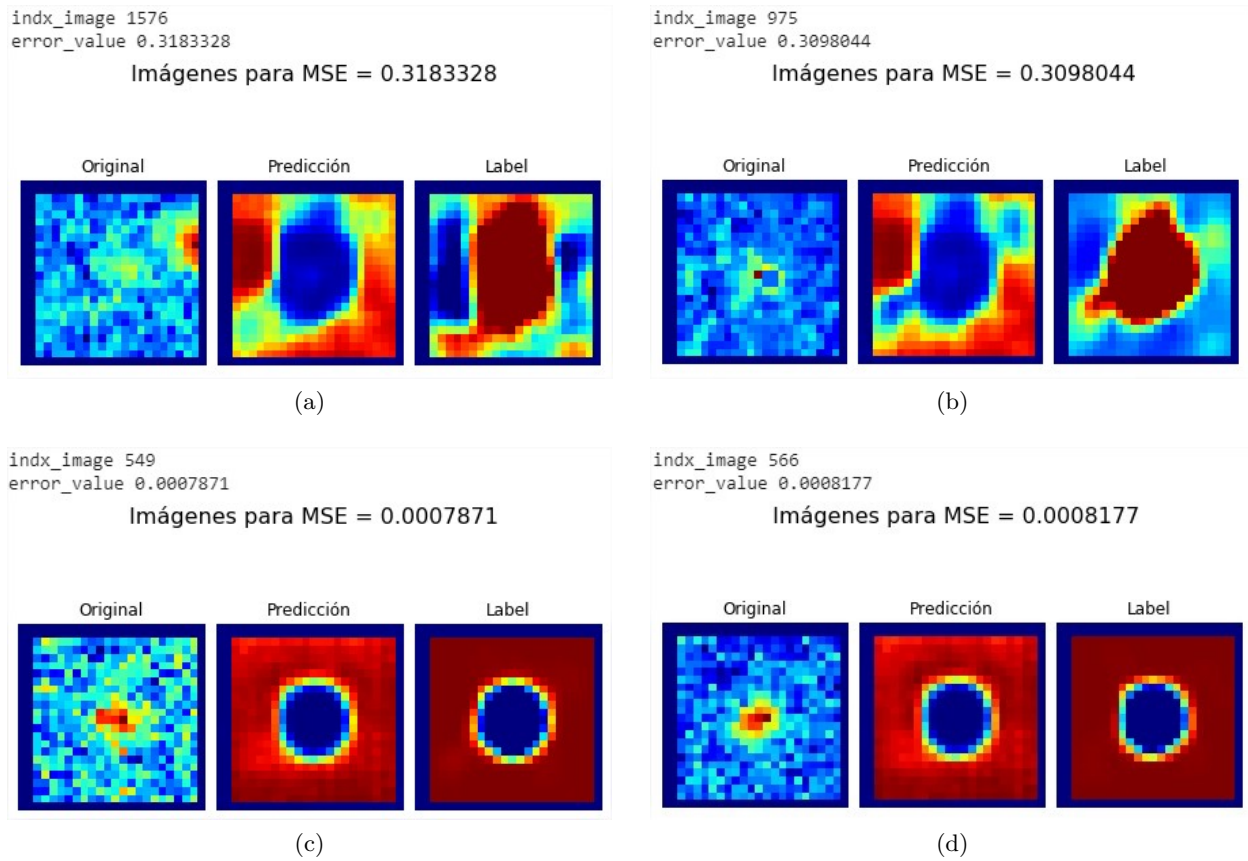


Figura 10: Ejemplos seleccionados correspondientes a modelo entrenado con Supernovas

### Experimento No.2

La Tabla 6 muestra los valores de MSE tanto de entrenamiento como de validación para ambos modelos, en la cual se evidencia que el modelo de autoencoder con la introducción de la característica codificada de la forma One-Hot obtiene un valor igual a 0.01052 en validación, lo que representa el menor valor de MSE de validación por lo que fue escogido como el mejor modelo de los dos entrenados.

La razón de la disminución del valor de MSE cuando la característica inyectada cambia de Numeral a One-hot encoded se debe a la disminución de la incertidumbre. Al agregar más certeza a las características y normalizadas como es el caso de One-hot encoding que agrega 2 características en lugar de una (Numeral). También debe tenerse en cuenta que se utilizó la función de activación "Softmax" que reparte a la capa densa probabilidades a cada característica y disminuyendo la incertidumbre se agrega más información a las otras características, esto porque se mantiene el número total de características.



Tabla 6: Valores de MSE para modelo concatenado con etiquetas Numeral y etiquetas One-hot encoded

Modelo	MSE train	MSE val
Numeral	0.01038	0.01204
One-Hot encoding	<b>0.00956</b>	<b>0.01052</b>

### Experimento No.3

Las Figura 11 muestra los valores de MSE de validación y de entrenamiento de los modelos resultantes de los cortes realizados al autoencoder original, donde es evidente el aumento de los errores con cada corte, por lo que el mejor modelo encontrado fue el modelo de autoencoder original, el cual alcanzó un valor de error de 0.01416 que representa el menor valor de MSE de validación.

Las Figuras 12 y 13 muestran las curvas de Loss pertenecientes al modelo de autoencoder original (mejor modelo), donde el valor de pérdida de entrenamiento alcanzado en la novena época fue de 0.3425 y el alcanzado en validación fue de 0.346 .

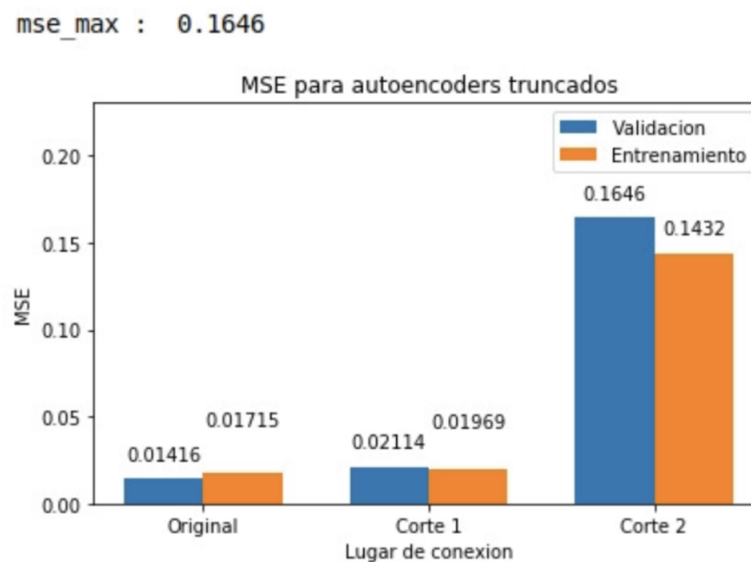


Figura 11: Resultados de MSE de validación y entrenamiento promedio para diferentes tipos de autoencoders

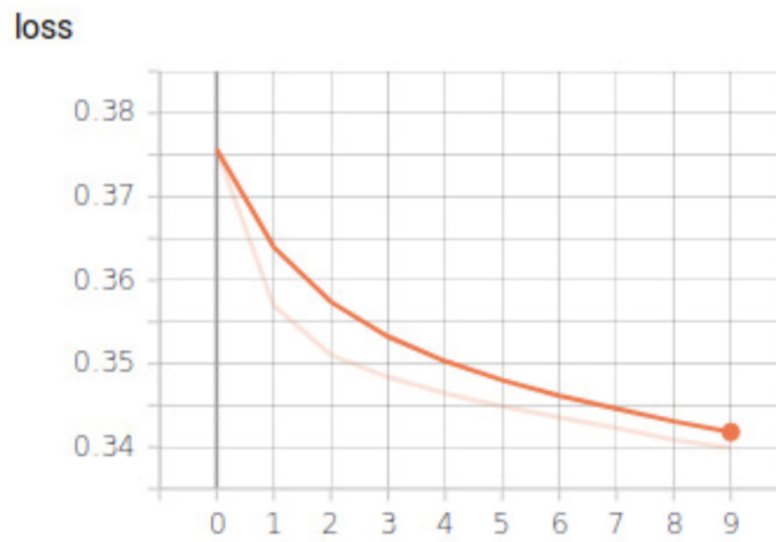


Figura 12: Resultados de Loss por epoca en entrenamiento para el mejor modelo del autoencoder

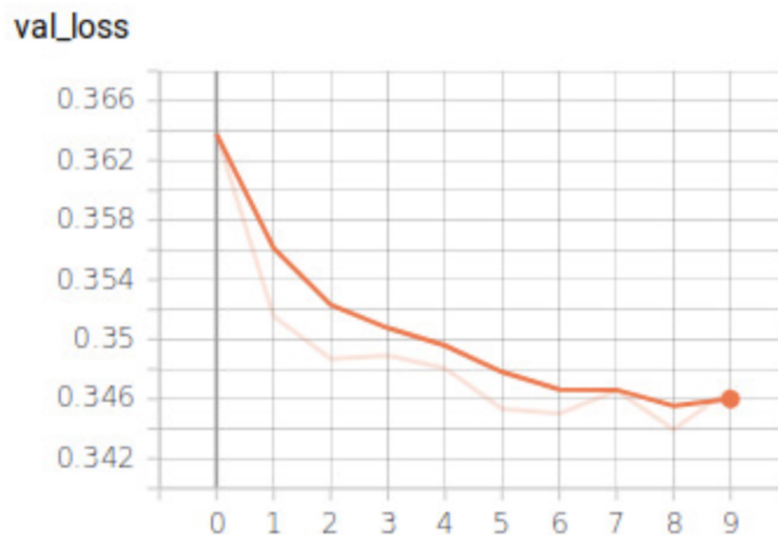


Figura 13: Resultados de Loss por epoca de validación para el mejor modelo del autoencoder

La Figura 11 muestra que a medida que se realizan cortes en el modelo autoencoder original el error aumenta porque cada vez se pierden más capas de convolución, las que son encargadas de extraer las características de la imagen. Por lo tanto, a medida que se pierden más capas convolucionales, a causa de los cortes, disminuye la capacidad de la red de extraer características de interés de las imágenes, por lo que el error aumenta.

### Experimento No.4

Las Tablas 7 y 8 muestran los valores de MSE de entrenamiento y validación correspondientes a los 5 entrenamientos realizados por modelo. La Tabla 9 muestra los valores medios y desviación estandar asociada a los 5 valores anteriores, respectivamente.

Los valores de desviación estándar calculados son muy similares para ambos modelos, sin embargo con respecto al valor promedio se obtiene un valor de 0.01309 para el modelo congelado y un valor de 0.00909 para el modelo condicionado, lo que evidencia una disminución de 0.004, por lo que el modelo condicionado será el escogido como el mejor modelo porque provoca en sus predicciones el menor valor de MSE como promedio .

Tabla 7: Muestras para modelo congelado

Modelo	BBDD	MSE train	MSE val
Congelado	Supernova	0.01267	0.01278
		0.01358	0.01353
		0.01159	0.01168
		0.01363	0.01371
		0.01144	0.01147

Tabla 8: Muestras para modelo condicionado

Modelo	BBDD	MSE train	MSE val
Congelado	Datos mixtos	0.00956	0.01052
		0.00872	0.00931
		0.00789	0.00689
		0.00905	0.00837
		0.00894	0.00871

Tabla 9: Estadísticas para comparar modelo de pesos congelados no condicionado y pesos congelados condicionado

ESTADISTICAS	MODELOS			
	Congelado		Condicionado	
	MSE train	MSE val	MSE train	MSE val
Valor medio	0.01299	0.01309	0.00908	<b>0.00909</b>
Desviación Estándar	0.00126	0.00132	0.00075	<b>0.00131</b>

## 4.2. Mejor modelo

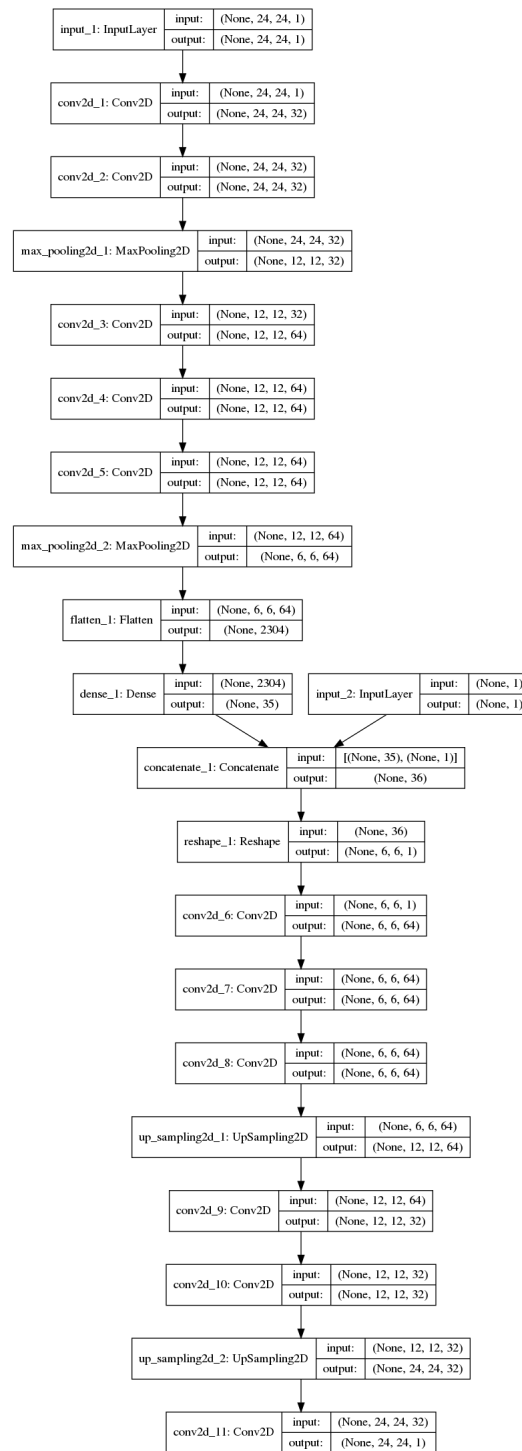


Figura 14: Arquitectura del mejor modelo de Autoencoder, con condicionamiento

## 5. Conclusiones

Fueron cumplidos los objetivos del proyecto porque se obtuvo un modelo autoencoder capaz de estimar los mapas de oclusión correspondientes a las imágenes de interés, el cual no solo redujo el tiempo de procesamiento y la complejidad inherente al cálculo de mapas de oclusión, de  $N \times N$  inferencias a 1, sino también que obtuvo el valor de MSE de validación más bajo de todas las propuestas sometidas a prueba.

La selección de la propuesta final fue resultado de la ejecución de 4 experimentos, a partir de los cuales se pueden tomar las siguientes ideas como conclusión:

1. La transferencia de pesos previamente entrenados en una red arbitraria y que hacen de esta un modelo eficiente es una buena opción para mejorar el rendimiento de otros modelos de interés.
2. La introducción de características en el cuello de botella del autoencoder, en especial la codificada mediante One-Hot, provocó una disminución del MSE.
3. Por otra parte, la reducción de capas en el autoencoder convolucional significó para el modelo la disminución de la capacidad de extraer características de las imágenes y el aumento del MSE.

## Referencias

- [1] *Occlusion experiments for image segmentation*,  
<https://github.com/akshaychawla/Occlusion-experiments-for-image-segmentation>
- [2] W.Badr, *Autoencoder: What is it? And what is it used for? (Parte 1)*,  
<https://towardsdatascience.com/auto-encoder-what-is-it-and-what-is-it-used-for-part-1-3e5c6f017726>
- [3] E.Dirk, *Bases de datos de imágenes astronómicas: HiTS2013*,  
[https://drive.google.com/drive/folders/1bjm8VnxkxrZIYpSLaOCLziyHuHngu3U\\_\\_](https://drive.google.com/drive/folders/1bjm8VnxkxrZIYpSLaOCLziyHuHngu3U__)
- [4] G. Cabrera-Vives, I. Reyes, F. Förster, P. A. Estévez, and J.-C. Maureira, “Deep-HiTS: Rotation Invariant Convolutional Neural Network for Transient Detection,” *Astrophys. J.*, vol. 836, no. 1, p. 97, 2017.
- [5] Repositorio del Proyecto, *Repositorio Github Proyecto EL4106*, <https://github.com/AndreTeixeira1998/EL4106-1oProyecto?files=1>
- [6] Pass two inputs in list to keras model build, <https://www.pyimagesearch.com/2019/02/04/keras-multiple-inputs-and-mixed-data/>
- [7] *Bases de datos de imágenes conformadas para experimentos: Images-No-Nova-Occlusion-20k-samples, Images-Nova-Occlusion-20k-samples y Images-Occlusion-40k-samples*,  
<https://drive.google.com/drive/u/0/folders/1Ix1JSHa6UAcWqaisbZcB2oWIwZzMNBT0>