

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ  
ESCUELA DE POSGRADO



**“PREDICCIÓN DE PUNTOS DE VISTA DE IMÁGENES 2D USANDO  
DEEP LEARNING”**

Trabajo de investigación para optar el grado de Magíster en  
Informática con mención en Ciencias de la Computación

AUTOR

Miguel Ángel Retamozo Sánchez

ASESOR

Ivan Anselmo Sipiran Mendoza

JURADO

Edwin Rafael Villanueva Talavera

Felix Arturo Oncevay Marcos

LIMA - PERÚ

2019

# Predicción de Puntos de Vista de Imágenes 2D Usando Deep Learning

Miguel Retamozo, Ivan Sipiran.  
Pontificia Universidad Católica del Perú.  
Departamento de Informatica - Ciencias de la Computación.  
{miguel.retamozo, isipiran}@pucp.edu.pe.

**Abstract**—El objetivo de este artículo es predecir los ángulos que describen los puntos de vista de objetos como (sillas, carros, sofás y televisores) en imágenes reales usando Deep Learning. Para lograrlo se desarrolló un renderizador de imágenes 2D en lenguaje C y se usó la API de OpenGL como librería gráfica el cual genera imágenes a partir de modelos de objetos 3D del repositorio ModelNet.

El presente artículo contribuye con dos métodos para renderizar las imágenes, el primero fue llamado *método esférico* ya que usa coordenadas esféricas para describir la rotación del objeto y consiste en mantener fijo el objeto 3D en el origen para trasladar la posición de la cámara describiendo círculos de diferente radio a lo largo del eje Z circunscritos en una esfera de radio R, el segundo fue llamado *método euleriano* ya que usa los ángulos de euler y consiste en mantener fija la posición de la cámara a una distancia R sobre el eje Z y rotar 3 veces por iteración respecto a los ejes Z, X' y Z". Se diseñó una red neuronal convolucional basada en la arquitectura de la red VGG la cual fue entrenada con imágenes generada por el renderizador.

## I. INTRODUCCIÓN

Desde que Joseph Nicéphore tomó la primera fotografía en el año 1826 con una primitiva cámara diseñada por él hasta la actualidad [1] el ser humano ha podido inmortalizar sus recuerdos en fotografías las cuales guardan información como luz, sombra, objetos, personas, plantas pero esta información dentro de una sola fotografía tiene en común el punto de vista (ViewPoint) desde el cual fue tomada la fotografía.

La importancia de los ángulos que describen el punto de vista de un objeto se puede evidenciar en robótica, ya que los robots modernos poseen algoritmos predictivos de puntos de vista los cuales les ayudan a saber como coger un objeto o saber por dónde o hacia donde ir [8]. Los software de reconstrucción de imágenes ya usan los puntos de vista para poder predecir profundidad en cada eje y así poder reconstruir un objeto 3D desde una simple imagen 2D.

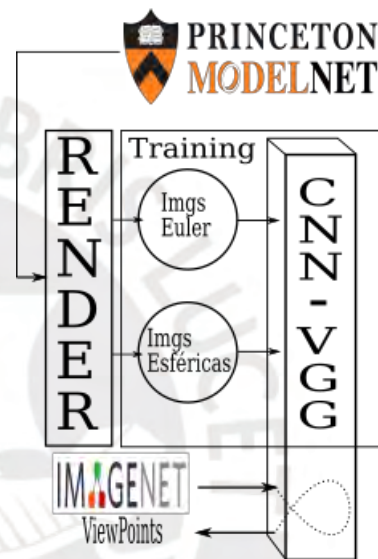


Fig. 1. Descripción general. Se renderiza imágenes de objetos 3D del repositorio de ModelNet con los métodos propuestos (esférico y euler) para entrenar una red neuronal convolucional de arquitectura VGG y así predecir los puntos de vista de objetos en imágenes reales de ImageNet del repositorio de PASCAL3D.

El enfoque de este trabajo se centra en predecir los puntos de vista de objetos en imágenes reales a partir de una red neuronal convolucional de arquitectura VGG entrenada con imágenes renderizadas de objetos 3D del repositorio de ModelNet con los métodos euler y esférico Figura 1.

## II. TRABAJOS RELACIONADOS

En la actualidad se ha demostrado que las redes neuronales convolucionales CNN superan a los métodos de piezas deformables (DPM) [3, 4, 16] en reconocimiento de objetos y esto es debido a que las CNN permite capturar partes de la imagen con convoluciones jerárquicas mientras que en las DPM realizan tareas para reconocer determinadas partes.

El dataset mas usado en los últimos años visto en los artículos [5, 6, 14, 15] para realizar tareas de clasificación de imágenes es el PASCALVOC, sin embargo para este trabajo se uso el repositorio de ModelNet debido a que se requiere Objetos 3D. Liang-Jun Zhang, Chao-Chen Gu y Kai-Jie Wu en Model-based Active Viewpoint Transfer for Purposive Perception [8] argumentan que para generar con mayor facilidad los puntos clave de una imagen se puede recurrir a los puntos de vista calculados con CNN, adicionalmente desarrollaron un modelo predictivo con deep learning para enseñarle a un brazo robótico a predecir los puntos de vista de una pieza metálica para así poder cogerlo. Se implementó un programa para tomar imágenes de las piezas en CAD con sus respectivos ángulos esféricos para poder entrenar una red convolucional.

Shubham Tulsiani y Jitendra Malik en Viewpoints and Keypoints [5] menciona que se puede usar modelos renderizados 3D para entrenar una red neuronal convolucional y con esto poder estimar los puntos de vista de imágenes reales.

### III. METODOLOGÍA

A continuación se describe el fundamento matemático usado en el renderizador para los métodos esférico y euler, las condiciones necesarias para que el renderizador no cree imágenes repetidas y la arquitectura de red convolucional usada.

#### 1. Método Esférico

Este método consiste en mantener el objeto fijo en el centro de coordenadas  $O$  y mover la posición de la cámara descrito por el punto  $P(\rho, \phi, \theta)$  haciendo círculos en la superficie de la esfera por niveles de  $Z$  como se puede ver en la Figura 2.

##### 1.1. Restricciones del Método.

- El ángulo  $\theta$  se mueve cada  $s_\theta$  grados por iteración en el plano descrito por el recorrido de  $\theta$ .
- El ángulo  $\phi$  se mueve cada  $s_\phi$  grados por iteración en el plano descrito por el recorrido de  $\phi$ .
- Sea  $N$  : Número total de imágenes.
- Sea XYZ los ejes del plano carteciano descritos en coordenadas esféricas :

$$x = \rho \sin(\theta) \cos(\phi); \quad (1)$$

$$y = \rho \sin(\theta) \sin(\phi); \quad (2)$$

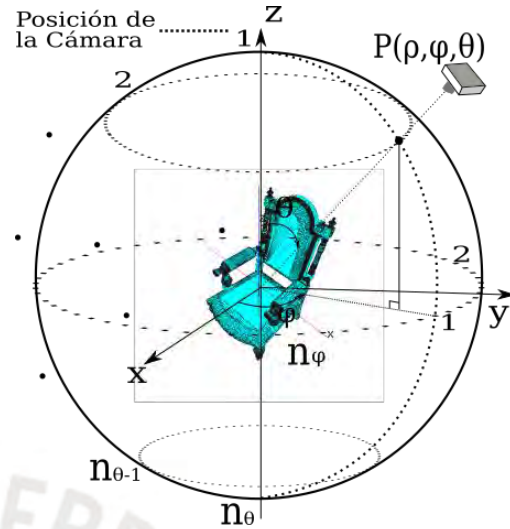


Fig. 2. Se muestra el recorrido de la cámara para el método esférico.

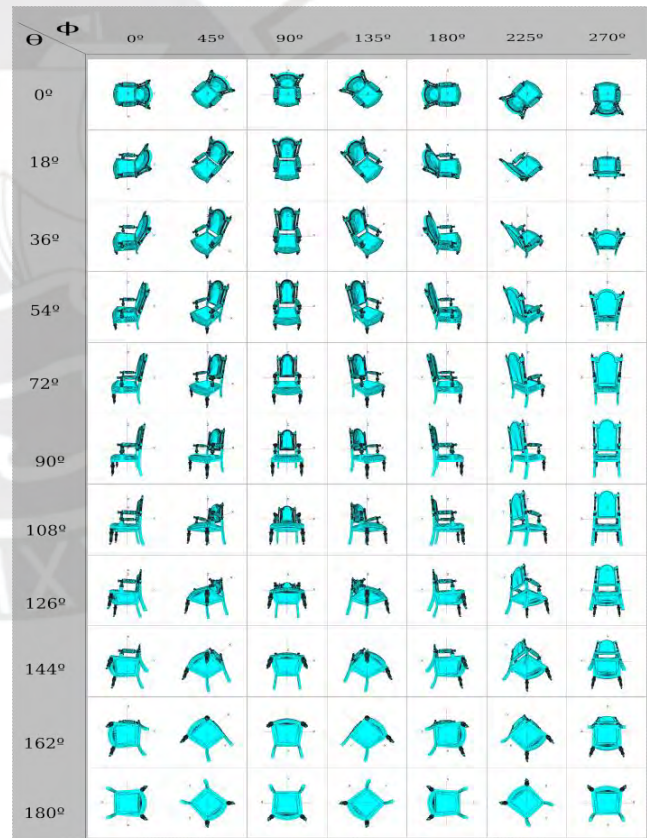


Fig. 3. Se muestran una porción de imágenes renderizadas con el método esférico para el objeto chair\_0058.off del repositorio de ModelNet.

$$z = \rho \cos(\theta) \quad (3)$$

$$0 \leq \theta; 0 \leq \phi < \pi \quad (4)$$

$$n_\theta = 180/s_\theta; n_\phi = 360/s_\phi \quad (5)$$

$$N = n_\theta \cdot n_\phi \quad (6)$$

Para el presente trabajo  $s_\theta = 9^\circ$  y  $s_\phi = 18^\circ$ , por lo tanto se generó 20 imágenes por nivel de  $\theta$  con un total de 20 niveles se renderizó 400 imágenes por objeto 3D, el resultado obtenido se puede ver en la Figura 3.

## 2. Método Euler

Este método consiste en mantener fijo la cámara y hacer rotar el objeto 3D describiendo los ángulos de euler ( $\phi$ ,  $\theta$ ,  $\psi$ ). Para lograrlo se hacen tres rotaciones consecutivas por iteración las cuales se describen respectivamente en las Figuras 4, 5 y 6.

De la Figura 4 el nuevo eje  $X'Y'Z'$  será descrito por :

$$\begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} = \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (7)$$

de (7) se puede escribir como :

$$E^J = Rz(\phi)E \quad (8)$$

De la Figura 5 el nuevo eje  $X''Y''Z''$  será descrito por :

$$\begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & \sin(\theta) \\ 0 & -\sin(\theta) & \cos(\theta) \end{bmatrix} \begin{bmatrix} X' \\ Y' \\ Z' \end{bmatrix} \quad (9)$$

de (9) se puede escribir como :

$$E^{JJ} = Rx'(\theta)E^J \quad (10)$$

De la Figura 6 el nuevo eje  $X'''Y'''Z'''$  será descrito por :

$$\begin{bmatrix} X''' \\ Y''' \\ Z''' \end{bmatrix} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X'' \\ Y'' \\ Z'' \end{bmatrix} \quad (11)$$

de (11) se puede escribir como :

$$E^{JJJ} = Rz''(\psi)E^{JJ} \quad (12)$$

de (8), (10) y (12)

$$E^{JJJ} = Rz(\phi)Rx'(\theta)Rz''(\psi)E \quad (13)$$

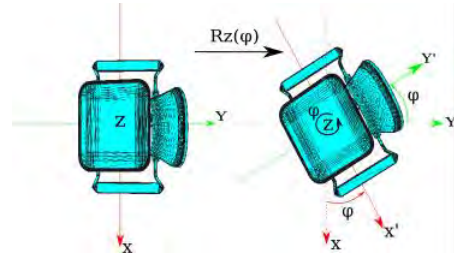


Fig. 4. Primera rotación,  $\phi$  grados con respecto al eje Z.

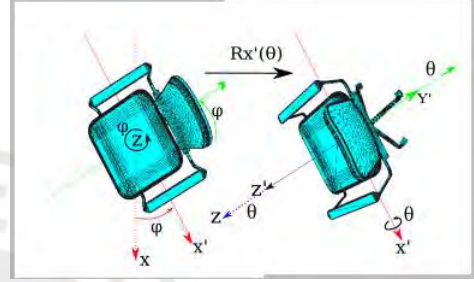


Fig. 5. Segunda rotación,  $\theta$  grados con respecto al eje  $X'$ .

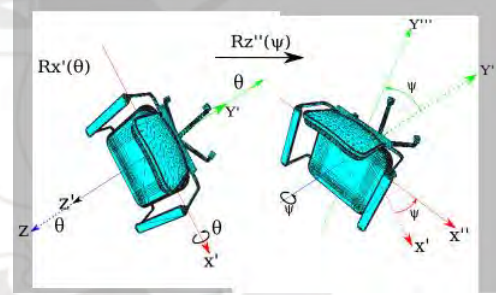


Fig. 6. Tercera rotación,  $\psi$  grados con respecto al eje  $Z''$ .

sea

$$R = Rz(\phi)Rx'(\theta)Rz''(\psi) \quad (14)$$

sea  $S(\phi) = \sin(\phi)$ ,  $C(\phi) = \cos(\phi)$  y desarrollando (14) se tiene la matriz de rotación de euler R en (15):

$$R = \begin{bmatrix} C(\psi)C(\phi) - C(\theta)S(\phi)S(\psi) & C(\psi)S(\phi) + C(\theta)C(\phi)S(\psi) & S(\theta)S(\phi) \\ S(\psi)C(\phi) - C(\theta)S(\phi)C(\psi) & S(\psi)S(\phi) + C(\theta)C(\phi)C(\psi) & S(\theta)C(\phi) \\ S(\theta)S(\phi) & -S(\theta)C(\phi) & C(\theta) \end{bmatrix} \quad (15)$$

### 2.1. Restricciones del Método de Euler

Para garantizar que las imágenes renderizadas no se repitan para el método de euler se tiene las siguientes consideraciones:

- El ángulo  $\phi$  se mueve cada  $s_\phi$  grados en el plano XY por iteración.

$$0 \leq \phi < 2\pi \quad (16)$$

$\psi$	$\theta$	$\phi$	40°	140°	240°	340°
0°	0°					
0°	18°					
0°	36°					
40°	54°					
40°	72°					
40°	90°					

Fig. 7. Se muestra una porción de imágenes renderizadas con el método de euler para el objeto chair\_0058 del repositorio de ModelNet.

- El ángulo  $\theta$  se mueve cada  $s_\theta$  grados en el plano  $X'Y'$  por iteración.

$$\theta \quad 0 \wedge 0 < \theta < 2\pi \quad (17)$$

- El ángulo  $\psi$  se mueve cada  $s_\psi$  grados en el plano  $X''Y''$  por iteración.

$$0 \leq \psi < 2\pi \quad (18)$$

- Sea  $N$  : Número total de imágenes.

$$n_\theta = 360/s_\theta; n_\phi = 180/s_\theta; n_\psi = 360/s_\psi \quad (19)$$

$$N = n_\theta n_\phi n_\psi \quad (20)$$

- Para nuestro caso

$$n_\theta = n_\phi = n_\psi = 36^\circ; N = 400 \quad (21)$$

### 3. Condiciones de Renderización

A continuacin se describiran las condiciones necesarias para renderizar correctamente las imágenes a partir de objetos 3D del repositorio de ModelNet.

#### 3.1. Posición Referencial Inicial.

El repositorio ModelNet[9] esta' compuesto por archivos con extension off los cuales contiene información de los vértices y la secuencia de cada vértice del objeto 3D, este artículo presenta como posición referencial inicial de cada objeto 3D las posiciones iniciales vistas en la Figura 8 la imagen izquierda es la original renderizada

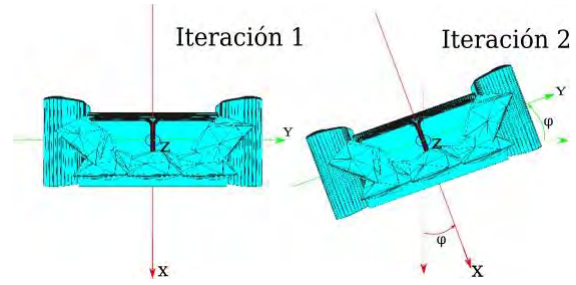


Fig. 8. Se muestra posiciones referenciales de la primera y segunda iteración de los objetos renderizados.

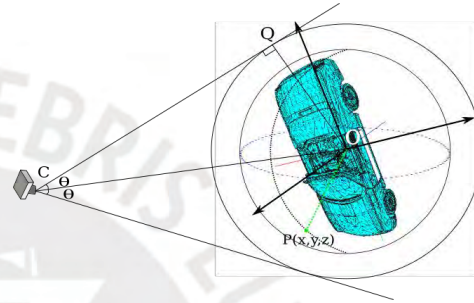


Fig. 9. Se muestra un objeto 3D en perspectiva circunscrito dentro de dos esferas, siendo la primera una esfera de radio  $OP$  y la segunda de radio  $OQ$ .

desde una posición  $P(0,0,Z)$ , donde  $Z > 0$ , la imagen de la derecha es la imagen rotada en la segunda iteración.

#### 3.2. Condiciones de Escalabilidad.

Como se puede observar en el modelo esférico el parámetro  $\rho$  no está siendo considerado por ser constante, en el modelo de rotaciones eulerianas la distancia de la cámara al objeto 3D también es constante por ende en la Figura 9 se plantea encontrar el punto  $P(x,y,z)$  que pertenezca a la superficie del objeto 3D y que posea mayor distancia euclidiana desde el centro de coordenadas  $O$ . En Figura 9 para el plano  $XYZ$  se crea una esfera de radio  $OP$  que circunscriba al objeto 3D en su totalidad por lo tanto :

$$\overline{OQ} > \overline{OP} \rightarrow \overline{OP} = k\overline{OQ} \quad (22)$$

Para nuestro caso en particular  $k=0.75$  y  $2\theta$  el ángulo de perspectiva, si se considera el objeto 3D esta centrado en  $O$  entonces  $OC$  es bisectriz de ángulo de perspectiva.

#### 4. Arquitectura de Red Convolutacional.

En la actualidad las Redes Neuronales Convolucionales (CNN) son las redes neuronales predilectas para realizar procesamiento de imágenes

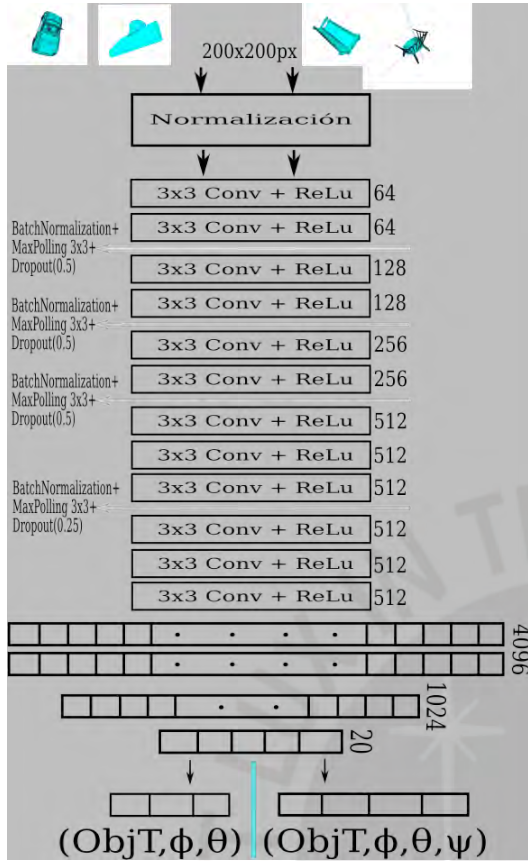


Fig. 10. Se muestra la arquitectura de la red CNN tipo VGG usada la cual posee 28M de parametros.

debido a sus principales componentes como la convolución y pooling los cuales en conjunto con los filtros dependiendo de la profundidad de las capas ayudan a extraer características simples y complejas de la imagen como ángulos, vértices y intersecciones. Al ser la presente investigación una predicción de ángulos de puntos de vista se uso regresión con una arquitectura basada en el modelo VGG[2], el cual consta de 12 capas convolucionales ocultas y 5 capas densas como se puede ver en la Figura 10.

#### 4.1. Función de Perdida.

Para este artículo de investigación se optó por usar MSE (Mean Squared Error ) como función de pérdida el cual esta comprendido de la siguiente manera :

$$L = \frac{1}{2m} \sum_{i=1}^m (f(x_i) - y_i)^2 \quad (23)$$

Donde  $y_i$  es una tupla de tamaño 4 (objType,  $\phi$ ,  $\theta$ ,  $\psi$ )

para el método de euler y tamaño 3 para el método esférico (objType,  $\phi$ ,  $\theta$ ) siendo objType el tipo de objeto para nuestro caso sillas, sofás, carros y televisores respectivamente.

## IV. EXPERIMENTACION

Se escogió el lenguaje de programación C y OpenGL como librería gráfica para desarrollar el renderizador de imágenes debido a sus bondades de alta velocidad de procesamiento y a la gran cantidad de imágenes necesarias para entrenar la red neuronal convolucional CNN-VGG, el programa desarrollado renderiza un archivo de extensión off [13] el cual contiene toda la información de los vértices y secuencias del modelo 3D. Para este artículo el renderizador genera 400 imagenes de 200 x 200 px en escala de grises por modelo 3D (archivo off). La CNN-VGG fue desarrollada en Python 3.37, se uso Keras[11] como framework y fue entrenada con 300 objetos 3D divididos en 4 categorías (75 sillas, 75 televisores, 75 sofás y 75 carros) diferentes escogidos aleatoriamente de un total de 1000 modelos del repositorio de ModelNet, esto quiere decir que se renderizo 120 000 imágenes, de las cuales se separó el 20% para validación y se entrenó con la diferencia ( 96000 imágenes), disminuyendo el valor de la función de pérdida entre 10% a 15% en poco más de 18 épocas para el método esférico y en poco mas de 34 épocas para el método de euler como se puede ver en las Figuras 11 y 12 respectivamente.

### 1. Predicción de Puntos de Vista.

Con la finalidad de validar el rendimiento de nuestros métodos propuestos se usó la distancia geodesica para ambos métodos, las cuales poseen diferente cálculo y se describen a continuación.

#### 1.1. Distancia Geodesica Método Esférico.

Al ser el cálculo sobre la superficie de una esfera se usó la fórmula de Vincenty para distancias sobre elipsoides con ejes mayores y menores iguales [17]. De acuerdo a la Figura 13 la distancia geodesica se describe a continuación:

$$\Delta\sigma = 2 \arcsin\left( \frac{\sqrt{S(\Delta\phi)^2 + C(\phi_p)C(\phi_q)S^2(\Delta\theta)}}{2} \right) \quad (24)$$

#### 1.2. Distancia Geodesica Método de Euler.

Propuesta por Tulsiani y Jitendra Malik en Viewpoints and Keypoints [5] la cual esta descrito por:

$$\Delta(R_1, R_2) = \frac{|\log(R_1^T R_2)|}{\sqrt{\frac{1}{2} F}} \quad (25)$$

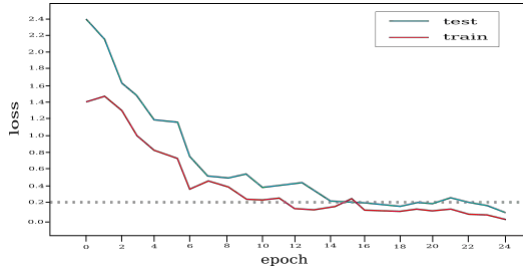


Fig. 11. El gráfico rojo y azul muestran el comportamiento de la función de pérdida con respecto a la cantidad de épocas en los procesos de entrenamiento y validación respectivamente para el **método esférico**.

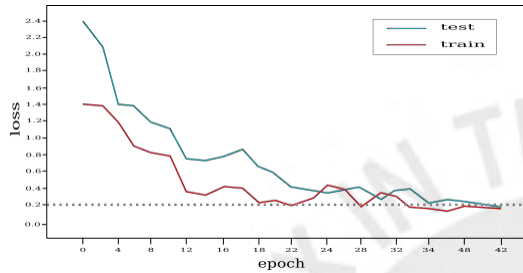


Fig. 12. El gráfico rojo y azul muestran el comportamiento de la función de pérdida con respecto a la cantidad de épocas en los procesos de entrenamiento y validación respectivamente para el **método de euler**.

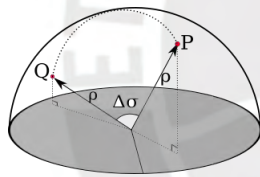


Fig. 13. Se muestra el ángulo central  $\Delta\sigma$  entre los puntos  $P$  y  $Q$  sobre la superficie de la esfera que forman el arco esférico  $PQ$ .

Siendo  $R$  la matriz de rotación de euler calculada en (15), para nuestro caso se tomó la distancia geodesica entre la matriz de rotación original y la matriz de rotación predicha  $\Delta(R_{pred}, R_{orig})$ .

### 1.3. Métrica

Con los resultados de las distancias geodesicas se usó la métrica Median Error o MedErr de acuerdo a los artículos [5,14] las predicciones de puntos de vista a menudo están alejadas y MedErr es una métrica de uso general que es robusta para estos casos.

## 2. Modelos Referenciales.

Se tomó como modelos referenciales los resultados obtenidos en los artículos [5,14] de MedErr, limitandonos para los objetos con que se entrenó

Author	Arquitectura	carro	silla	sofa	tv	avg
Tulsiani [5]	AlexNet[16]	8.8	14.8	13.7	15.4	13.2
Hao Su [14]	AlexNet[16]	<b>6.0</b>	<b>9.3</b>	<b>9.5</b>	12.6	<b>9.4</b>
M.Esférico	VGG[2]	10.4	11.3	10.2	<b>11.4</b>	10.8
M.Euler	VGG[2]	10.3	15.2	12.9	13.4	12.9

Table I. Evaluación de la métrica MedErr sobre la distancia geodesica basado en el modelo de Tulsiani, Malik [5], mientras menor su valor sera mejor.

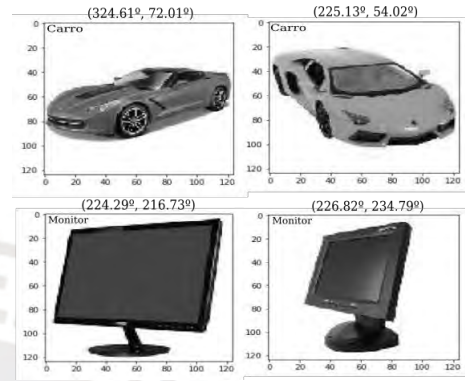


Fig. 14. Se ve los resultados obtenidos para imágenes clasificadas como sillas de imágenes reales de ImageNet del repositorio de PASCAL3D[12], aplicando el modelo predictivo con el método de rotaciones eulerianas.



Fig. 15. Se ve los resultados obtenidos para objetos clasificados como Carros y televisores de imágenes reales de ImageNet del repositorio PASCAL3D[12], usando 25 imágenes por categoría los resultados se puede observar en la Tabla 1.

la red (carros, sillas, sofás y televisores). Para poder probar la red entrenada se usó imágenes reales y sus respectivos objetos .mat de ImageNet del repositorio de PASCAL3D[12], usando 25 imágenes por categoría los resultados se puede observar en la Tabla 1.

A continuación en las Figuras 14 y 15 se muestra la predicción de imágenes reales de ImageNet del repositorio de PASCAL3D[12] escogidas aleatoriamente para los métodos de euler y esférico.

## V. CONCLUSIONES

En las condiciones en que se llevó a cabo la presente investigación se logró predecir los puntos de vista para los dos métodos de renderización propuestos con una CNN basada en la arquitectura VGG alcanzando a los métodos existentes en 4 clases de objetos de ImageNet en PASCAL3D.

Con los resultados obtenidos en la Tabla I y en las Figuras 11 y 12 podemos mencionar que el método esférico obtuvo mejores resultados que el método de Euler, dando a notar que usar puntos de vista en coordenadas esféricas reduce la complejidad computacional, cabe mencionar que no es posible aplicar dicho método a aviones o cualquier objeto con ángulo de elevación positivo.

Para un trabajo futuro se podría ampliar el conjunto de entrenamiento aumentando las clases de los objetos, aumentar la cantidad de canales por imagen ya que el modelo solo fue entrenado con imágenes en escala de grises.

## REFERENCES

- [1] Martín Soria y Alfredo Romero, El Honrado Inventor de la Fotografía, pp 1-4.
- [2] Karen Simonyan y Andrew Zisserman, Visual Geometry Group, Department of Engineering Science, University of Oxford, VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION in 2015, pp 3-12.
- [3] Ross Girshick, Forrest Iandola, Trevor Darrell y Jitendra Malik, Deformable Part Models are Convolutional Neural Networks 2015 . pp 437-439.
- [4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramana, Object detection with discriminatively trained part based models.
- [5] Shubham Tulsiani and Jitendra Malik, Viewpoints and Keypoints, 2015, 1510-1519 pp.
- [6] Yu Xiang, Roozbeh Mottaghi, Silvio Savarese, A Benchmark for 3D Object Detection in the Wild. 2014. pp 1-3.
- [7] Vincenty, Thaddeus (1975-04-01). "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations" (PDF). Survey Review. Kingston Road, Tolworth, Surrey: Directorate of Overseas Surveys. 23 (176): 8893. doi:10.1179/sre.1975.23.176.88. Retrieved 2008-07-21.
- [8] Liang-Jun Zhang, Chao-Chen Gu, Kai-Jie Wu, Model-based Active Viewpoint Transfer for Purposive Perception in 2017 13th IEEE Conference on Automation Science and Engineering (CASE)
- [9] The Princeton ModelNet. <http://modelnet.cs.princeton.edu/>. [Online; accessed March 2015]. 5
- [10] H. Su, M. Savva, L. Yi, A. X. Chang, S. Song, F. Yu, Z. Li, J. Xiao, Q. Huang, S. Savarese, T. Funkhouser, P. Hanrahan, and L. J. Guibas. ShapeNet: An information-rich 3d model repository. <http://www.shapenet.org/>. 2015.
- [11] Chollet, Fran y otros, chollet2015keras, Keras, <https://github.com/fchollet/keras>. 2015.
- [12] Y. Xiang, R. Mottaghi, and S. Savarese. Beyond pascal: A benchmark for 3d object detection in the wild. In Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on, pages 7582. IEEE, 2014.
- [13] OFF file format (Princeton), [http://segeval.cs.princeton.edu/public/off\\_format.html](http://segeval.cs.princeton.edu/public/off_format.html).
- [14] Hao Su, Charles R. Qi, Yangyan Li, Leonidas J. Guibas. Render for CNN: Viewpoint Estimation in Images Using CNNs Trained with Rendered 3D Model Views, 2015 IEEE International Conference on Computer Vision (ICCV), pp 2686-2694.
- [15] Arsalan Mousavian, Dragomir Anguelov, John Flynn, Jana Koecká, 3D Bounding Box Estimation Using Deep Learning and Geometry, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 5632-5640.
- [16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, and K. Weinberger, editors, Advances in Neural Information Processing Systems 25, pages 1097-1105. Curran Associates, Inc., 2012.
- [17] Vincenty, Thaddeus (1975-04-01). "Direct and Inverse Solutions of Geodesics on the Ellipsoid with Application of Nested Equations" (PDF). Survey Review. Kingston Road, Tolworth, Surrey: Directorate of Overseas Surveys. 23 (176): 8893. doi:10.1179/sre.1975.23.176.88. Retrieved 2008-07-21.