

Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ingeniería de Transporte y Logística



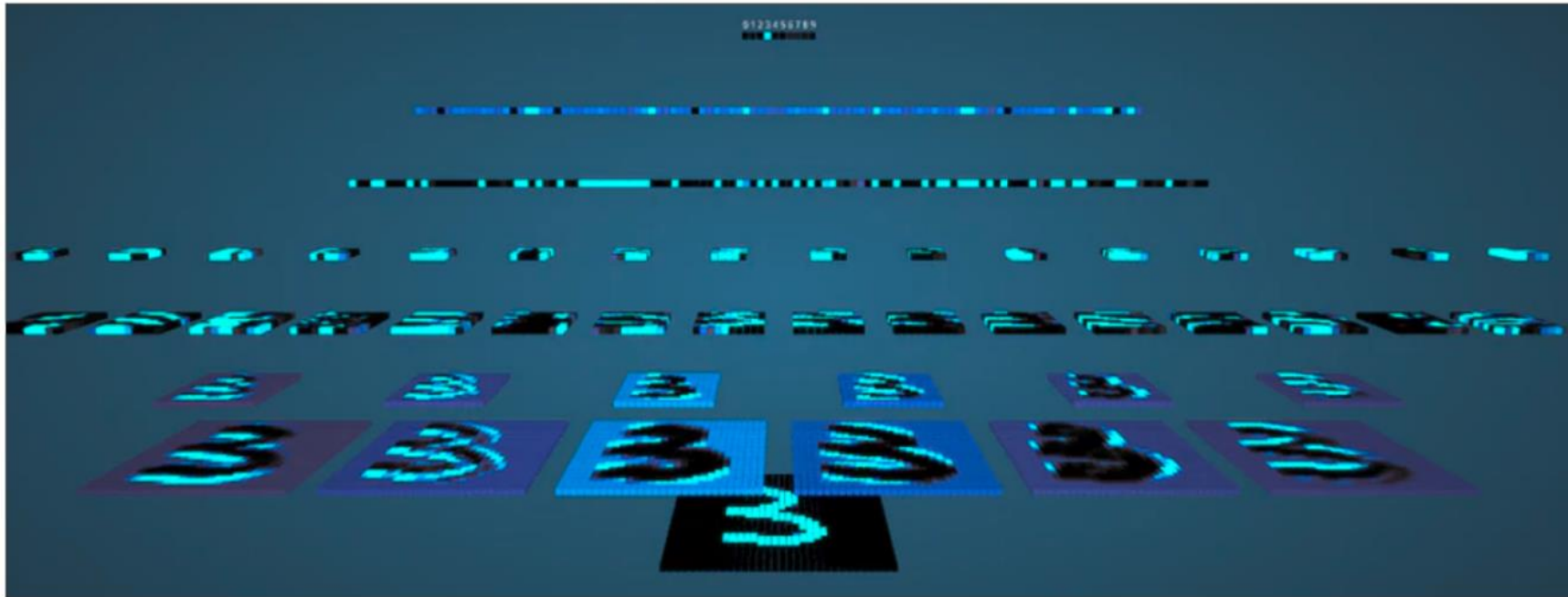
# Sistemas Urbanos Inteligentes

Aspectos prácticos de CNNs

Hans Löbel

Dpto. Ingeniería de Transporte y Logística  
Dpto. Ciencia de la Computación

Revisemos como se **ve** todo esto en la práctica



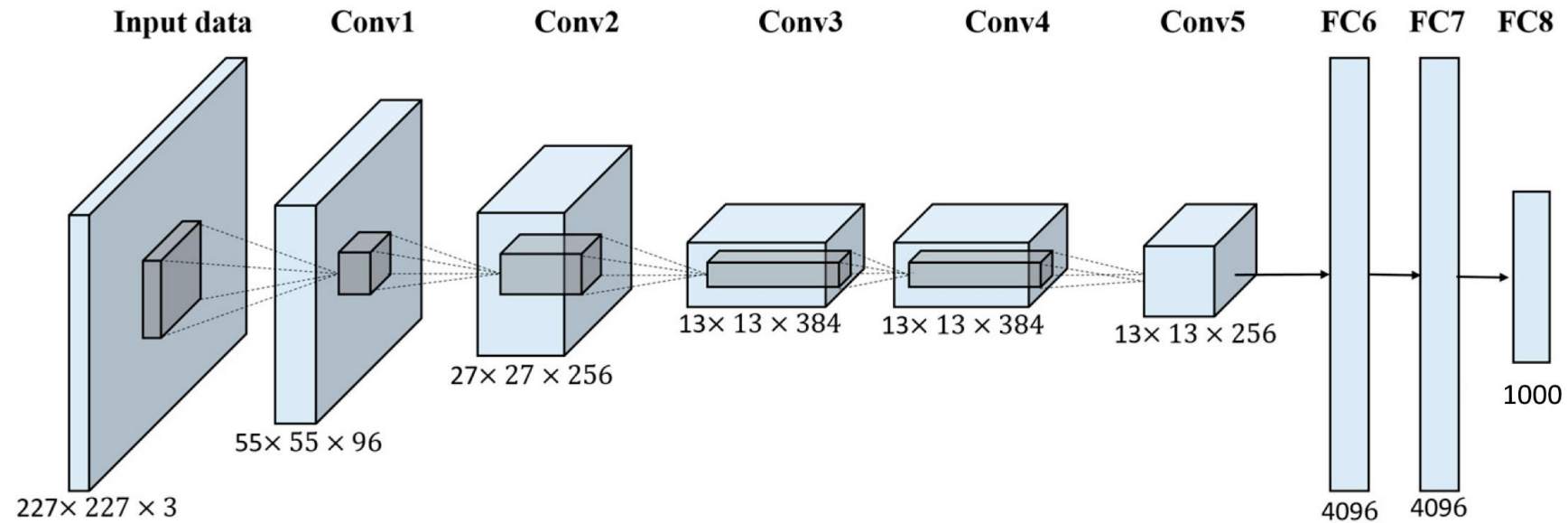
[www.cs.cmu.edu/~aharley/vis/conv/flat.html](http://www.cs.cmu.edu/~aharley/vis/conv/flat.html)

Veamos ahora un caso de estudio

# Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction

X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, Y. Wang (Sensors, 2017)

## AlexNet fue uno de los primeros ejemplos exitosos de ConvNets

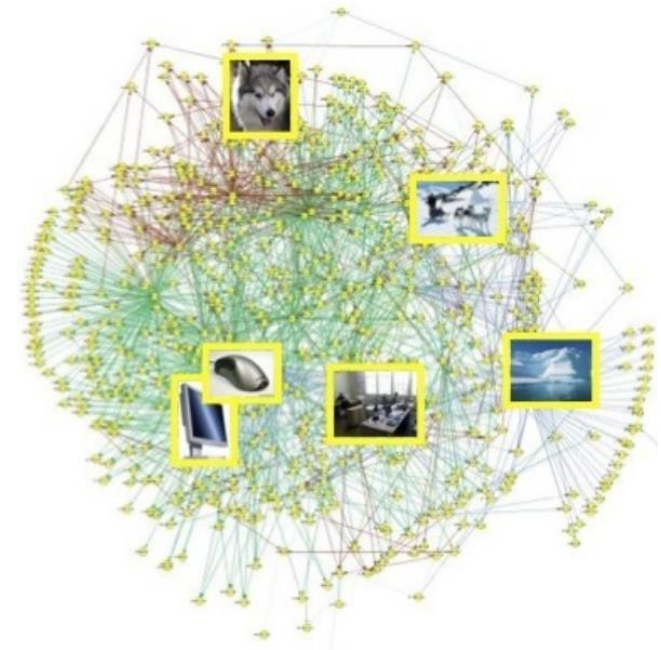


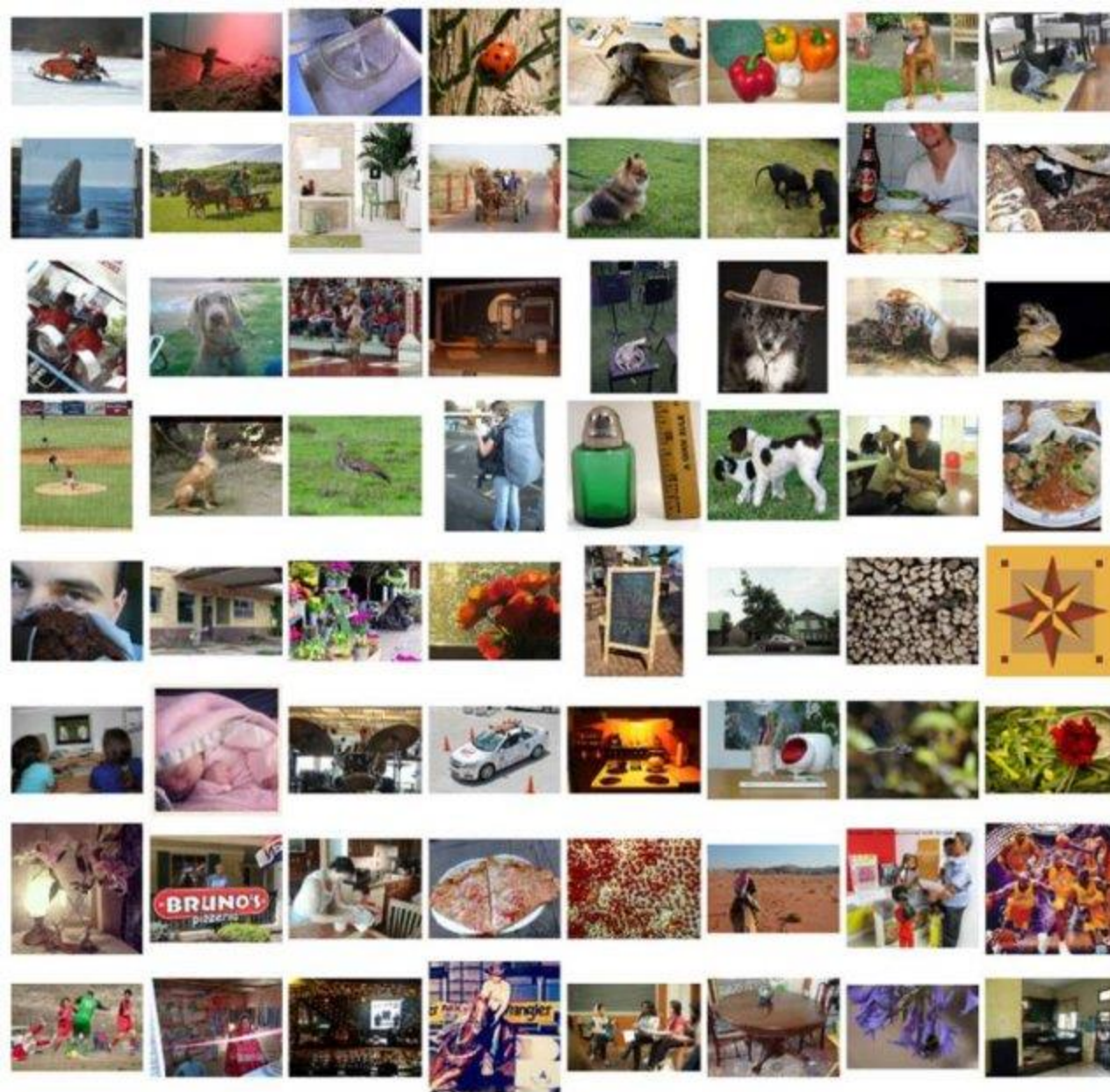
- Además de utilizar redes convoluciones, introdujo una serie de “trucos” muy útiles para el entrenamiento de redes (**ReLU**, **dropout**, **data augmentation**).
- Sin embargo, el gran diferenciador fue el ser capaz de aprender efectivamente de un set de datos órdenes de magnitud mayor que los existentes hasta ese momento.
- Esto abrió la puerta a un gran número de aplicaciones en distintos dominios, que vieron a las imágenes como un esquema útil de representación.

El set de datos ImageNet fue uno de los factores clave

- ImageNet es la base para la gran mayoría de los trabajos iniciales redes convolucionales.
- Este set de datos presenta más de 15M de imágenes en colores divididas en 22K categorías.
- Las imágenes fueron recolectadas de la web y rotuladas utilizando Amazon Mechanical Turk.
- Comúnmente se usa un subconjunto de ImageNet con 1K categorías y 1.2M de imágenes de entrenamiento (1.2K por categoría).

IMAGENET



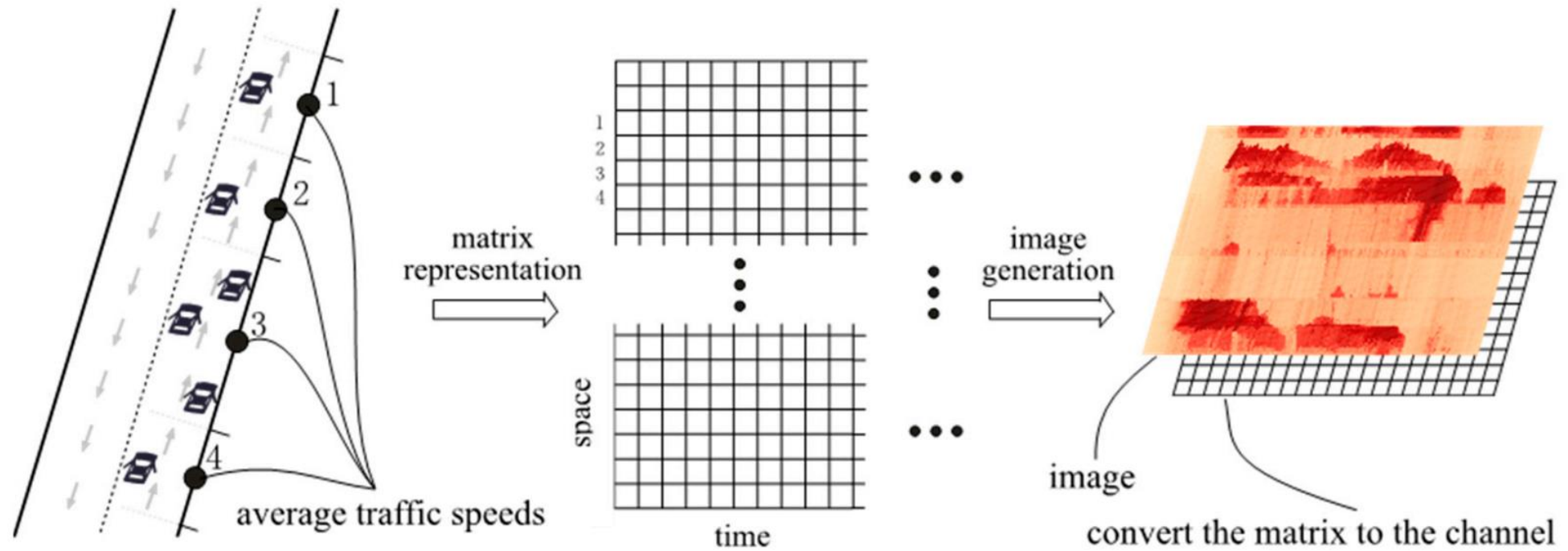


El problema es que el usar imágenes sigue siendo restrictivo!!!!

(no todo es imágenes/videos/audio/etc)



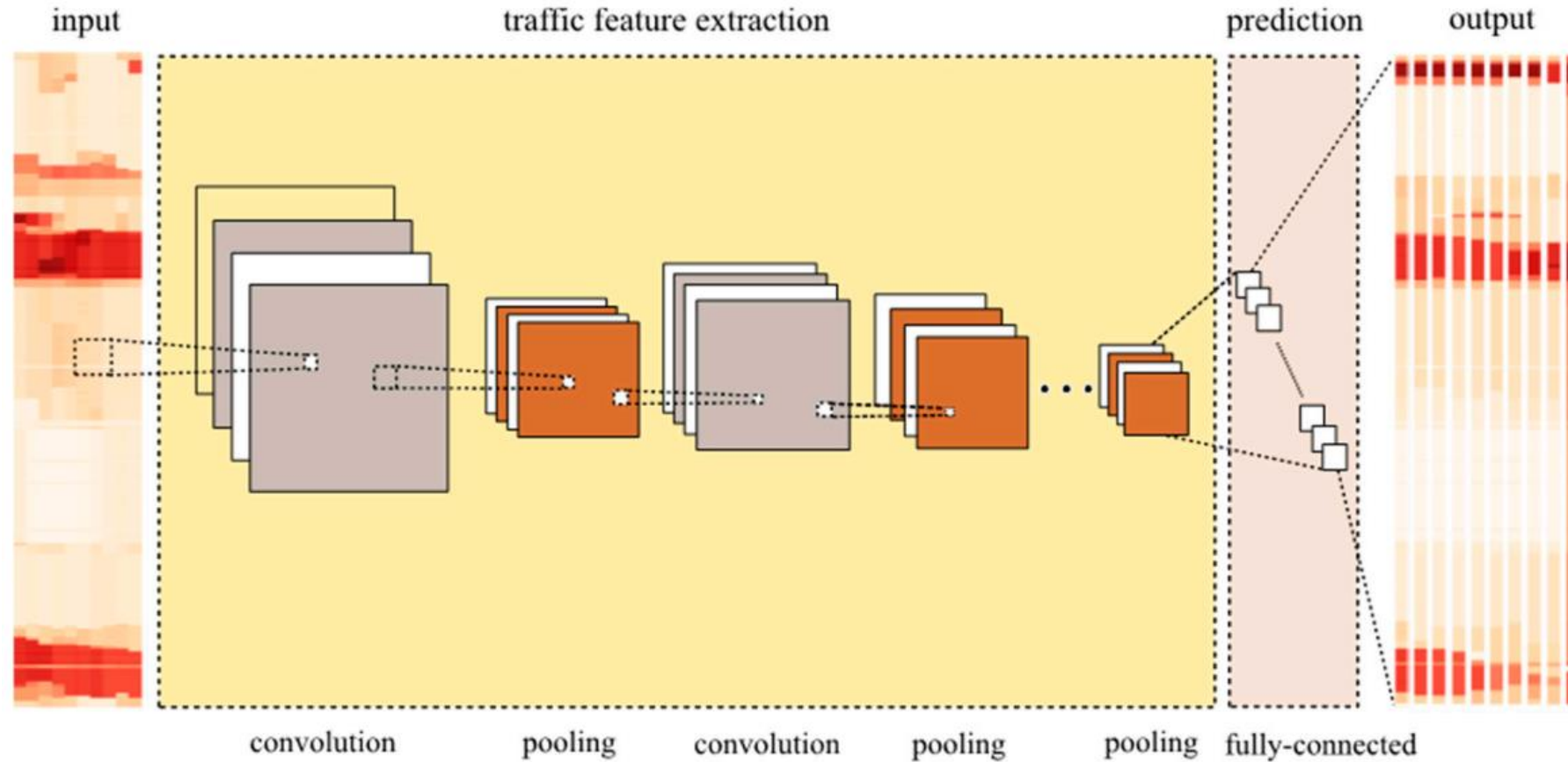
*Insight 1*: entradas a una CNN no tienen por qué ser *imágenes reales*



**Figure 1.** An illustration of the traffic-to-image conversion on a network.



Arquitectura es conceptualmente similar a la de AlexNet



**Figure 2.** Deep learning architecture of CNN in the context of transportation.

Arquitectura es conceptualmente similar a la de AlexNet

**Table 2.** Hyperparameters of the CNN.

Layer	Name	Parameters	Dimensions	Parameter Scale
Input	—	—	(1, 236, 20)	—
Layer 1	Convolution	Filter (256, 3, 3)	(256, 236, 20)	2304
Layer 1	Pooling	Pooling (2, 2)	(256, 118, 10)	0
Layer 2	Convolution	Filter (128, 3, 3)	(128, 118, 10)	1152
Layer 2	Pooling	Pooling (2, 2)	(128, 59, 5)	0
Layer 3	Convolution	Filter (64, 3, 3)	(64, 59, 5)	576
Layer 3	Pooling	Pooling (2, 2)	(64, 30, 3)	0
Layer 4	Data flatten	—	(5760, )	0
Layer 4	Fully-connected	—	(1180, )	6,796,800
Output	—	—	(1180, )	—

## Veamos los datos

- Registros de 10k taxis en Beijing, recolectados durante algo más de un mes en 2015.
- Datos agregados en intervalos de 2 minutos.
- 4 tareas de predicción: próximos 10 o 20 minutos, dados últimos 30 o 40 minutos.



Veamos los datos

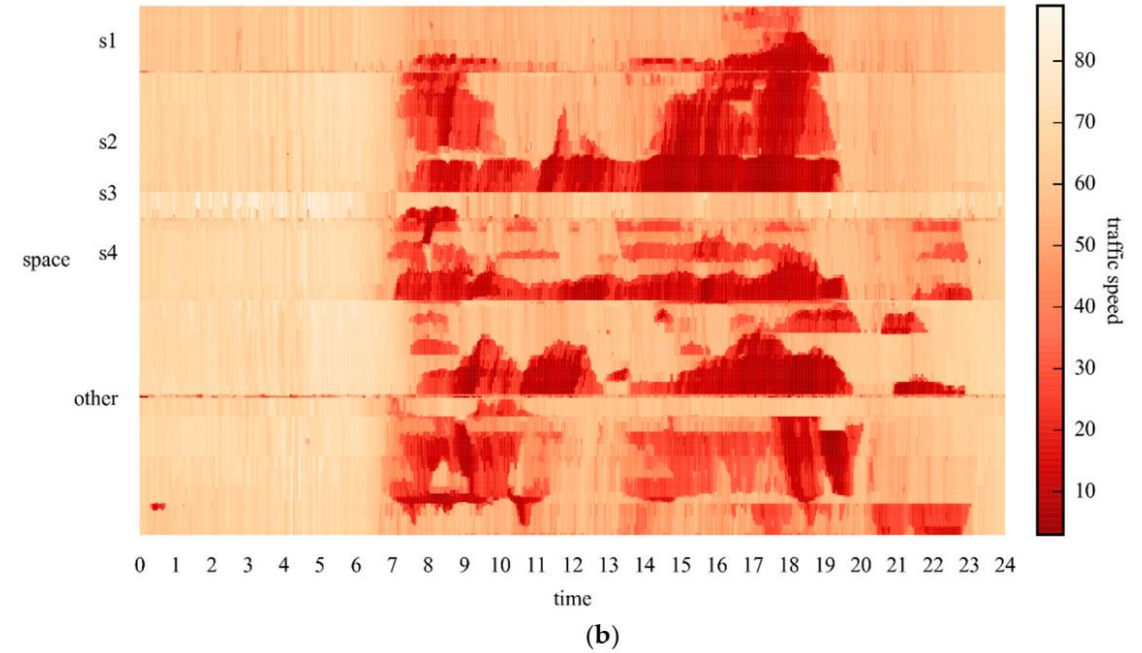
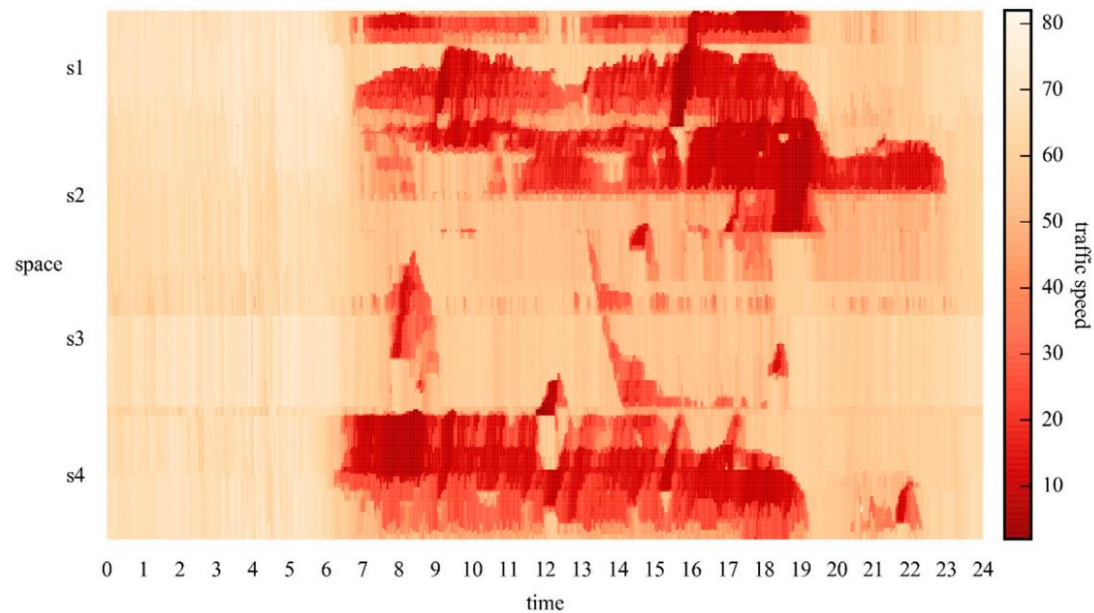
236 secciones



352 secciones



## Veamos los datos



- 37 imágenes fueron generadas para cada una de las redes (1 por día).
- Dependiendo de la tarea, se generan distintos sets de entrenamiento (primeros 30 días) y validación (últimos 7 días).
- Por ejemplo, para la tarea 40-10, se generan 21,6K ejemplos de entrenamiento y aprox. 5K de validación.

## Veamos algunos resultados

**Table 3.** Prediction performance (MSE) of the CNN and other algorithms.

Study Network	Model	MSE of Different Models (on Test Datasets)			
		Task 1	Task 2	Task 3	Task 4
Network 1	CNN	<b>22.825 *</b>	<b>24.345 *</b>	<b>30.593 *</b>	<b>31.424 *</b>
	OLS	27.047	31.273	41.334	48.107
	KNN	51.700	55.708	60.256	64.132
	RF	35.092	35.431	40.476	40.638
	ANN	67.764	52.339	58.797	57.225
	SAE	60.751	69.082	65.292	68.326
	RNN	33.408	36.833	40.551	39.038
	LSTM NN	37.759	33.218	42.909	42.865
Network 2	CNN	<b>27.163 *</b>	<b>28.479 *</b>	<b>37.987 *</b>	<b>38.816 *</b>
	OLS	33.741	41.657	50.123	62.282
	KNN	69.965	74.863	79.367	83.881
	RF	48.603	48.946	52.676	53.067
	ANN	124.937	147.489	133.299	168.136
	SAE	85.079	94.982	82.271	99.020
	RNN	48.877	47.470	52.577	52.114
	LSTM NN	43.304	45.657	50.928	48.345

Note: \* indicates the best result.



## Veamos algunos resultados

**Table 4.** Prediction performance (accuracy) of the CNN and other algorithms.

Study Network	Model	Accuracy Score of Different Models (on Test Datasets)			
		Task 1	Task 2	Task 3	Task 4
Network 1	CNN	<b>0.939 *</b>	<b>0.942 *</b>	<b>0.925 *</b>	<b>0.928 *</b>
	OLS	0.935	0.929	0.915	0.909
	KNN	0.901	0.897	0.893	0.890
	RF	0.917	0.917	0.910	0.910
	ANN	0.869	0.876	0.852	0.865
	SAE	0.867	0.870	0.866	0.866
	RNN	0.908	0.913	0.898	0.900
	LSTM NN	0.910	0.908	0.901	0.905
Network 2	CNN	<b>0.938 *</b>	<b>0.936 *</b>	<b>0.920 *</b>	<b>0.922 *</b>
	OLS	0.929	0.920	0.907	0.897
	KNN	0.886	0.884	0.879	0.876
	RF	0.898	0.898	0.893	0.892
	ANN	0.794	0.867	0.823	0.832
	SAE	0.846	0.835	0.848	0.825
	RNN	0.901	0.900	0.896	0.896
	LSTM NN	0.903	0.907	0.901	0.895

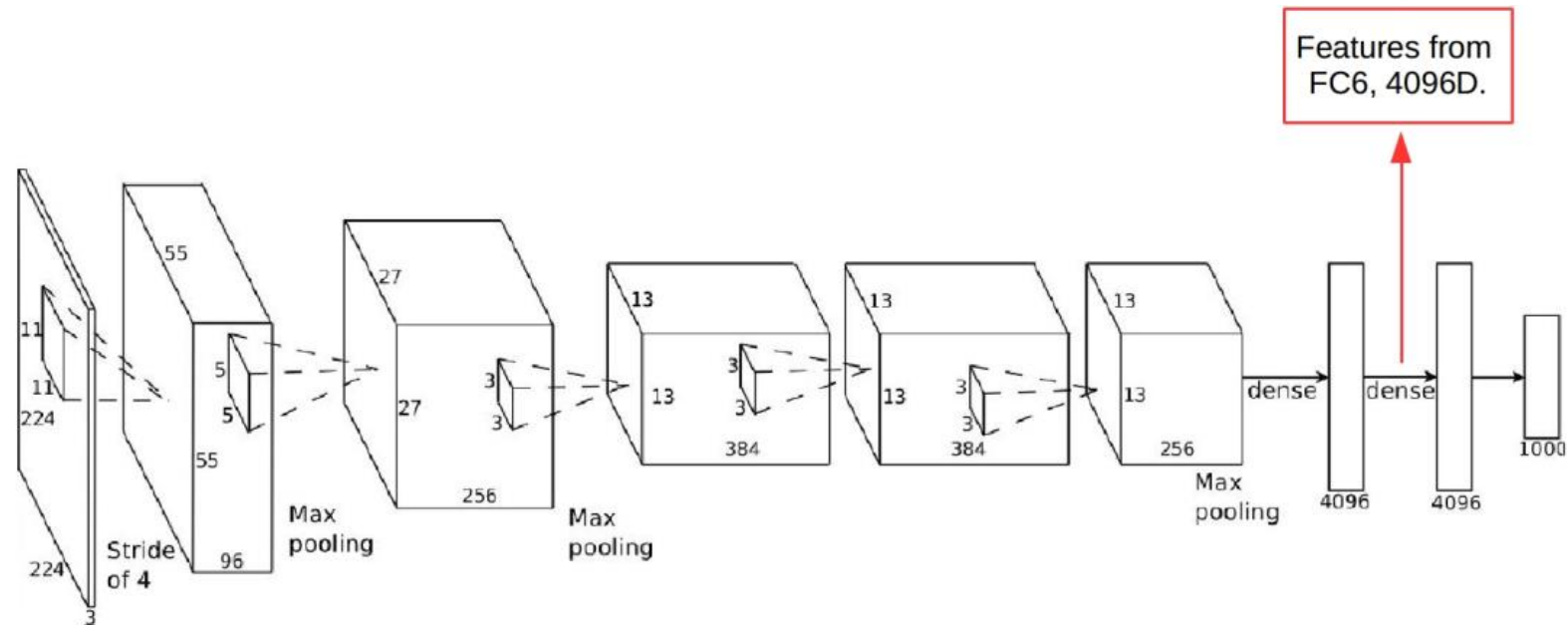
Note: \* indicates the best result.

Súper impresionante, pero sigue sin ser evidente como puedo aprovecharlo

- Estos resultados son claramente impresionantes, pero presentan un problema muy puntual, en una ciudad muy puntual.
- En particular, dada la cantidad de parámetros de estas redes, conseguir el volumen de datos necesario para entrenarlas sin caer en **overfitting** es difícil.
- Afortunadamente, las CNN poseen características que permiten enfrentar este problema de manera bastante intuitiva...

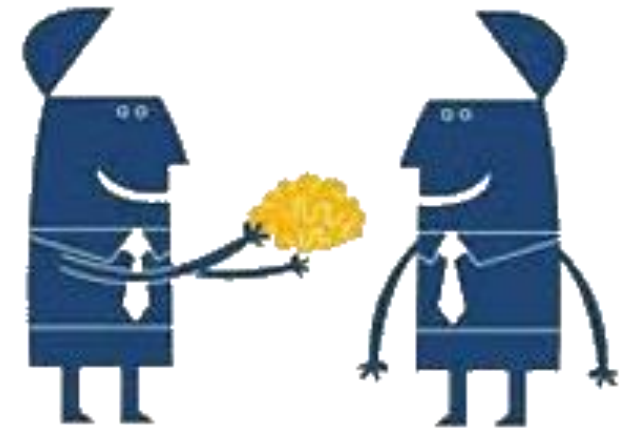
*Insight 2:* CNNs ya entrenadas pueden usarse en otros dominios, con poco o incluso sin un nuevo entrenamiento

- Un simple ejemplo podemos verlo con AlexNet.
- Si le “cortamos la cabeza” y la utilizamos como extractor de características, se obtienen excelentes resultados en una gran cantidad de tareas visuales: reconocimiento de escenas, caras, acciones, personas, etc.



Este tipo de ideas son parte de un área llamada *Domain Adaptation* (o *Transfer Learning*)

- En resumen, la idea es aprender inicialmente modelos poderosos utilizando sets de datos abundantes y de buena calidad.
- Luego, se transfiere de alguna manera el aprendizaje a un dominio (conjunto de datos) más limitado, ya sea en cuanto a cantidad, calidad o seguridad.
- Cómo y cuándo son temas de profunda investigación y con gran relevancia en muchos problemas, como por ejemplo el desarrollo de vehículos autónomos.



Este tipo de ideas son parte de un área llamada  
*Domain Adaptation* (o *Transfer Learning*)

### Source Domain

- Lots of labeled data

$$P_S = (X_S, Y_S)$$



### Target Domain

- Limited labels

$$P_T = (X_T, Y_T)$$



**Hope is that:**

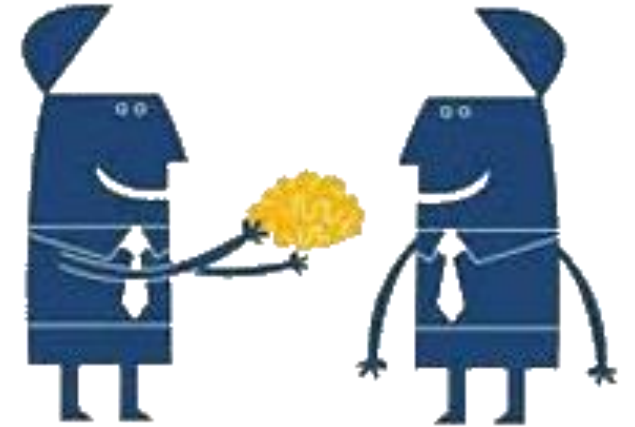
$$P_T \approx P_S$$

**Or at least**

$$P_T(X_T) \approx P_S(X_S)$$

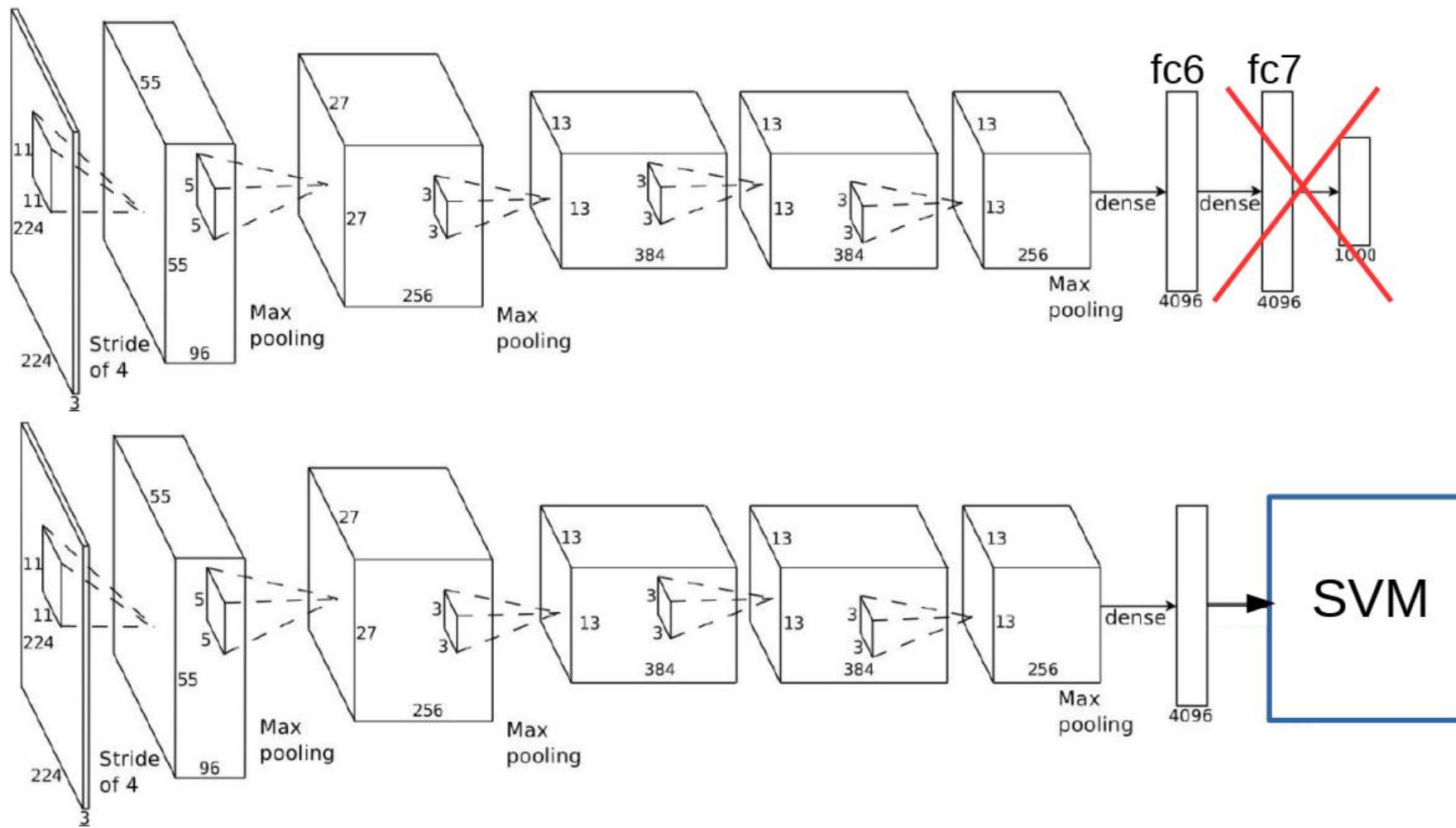
A nivel básico, existen dos mecanismos típicos para hacer esta adaptación

- **Transferencia directa:** features aprendidas en un dominio se utilizan directamente en otro (solo se cambia el clasificador).
- **Finetuning:** las features aprendidas son reentrenadas “suavemente” en el nuevo dominio.



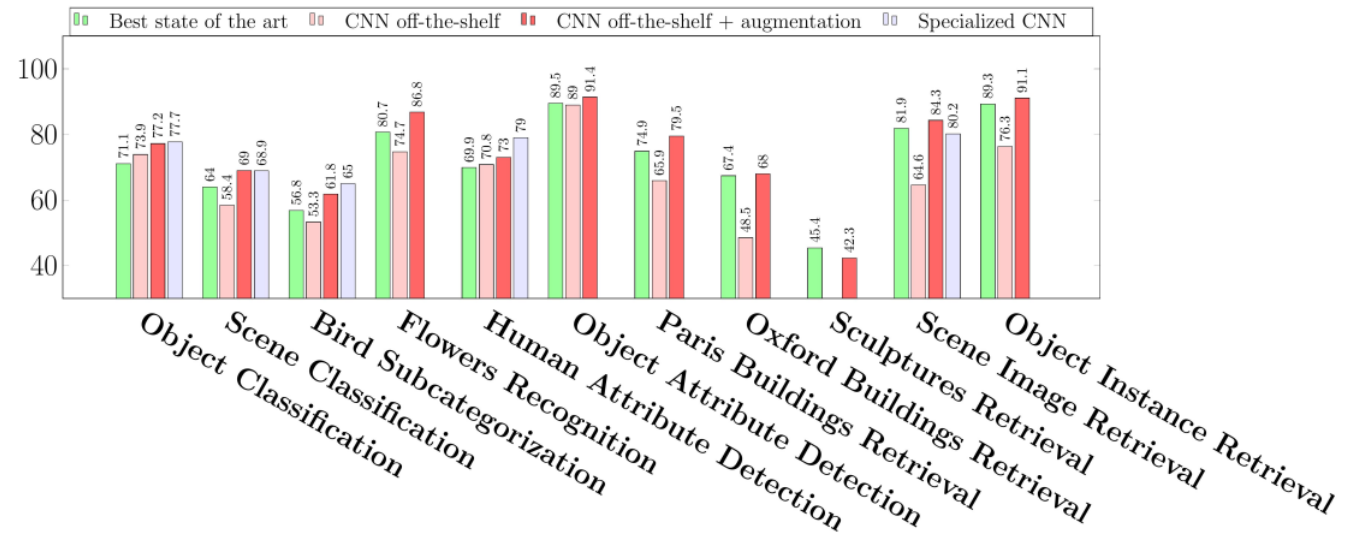
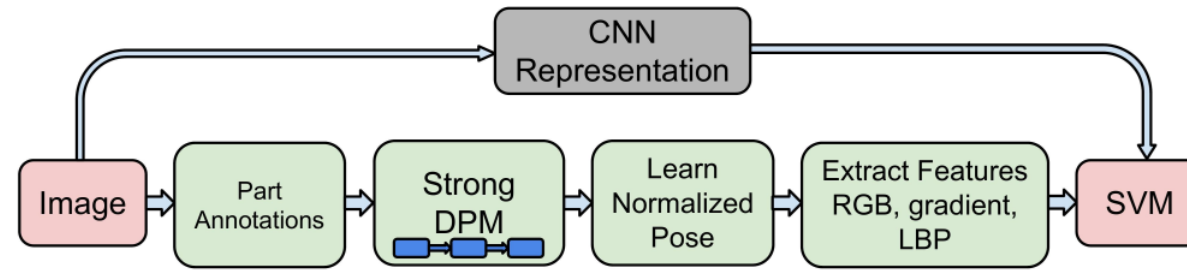


## Transferencia directa de features

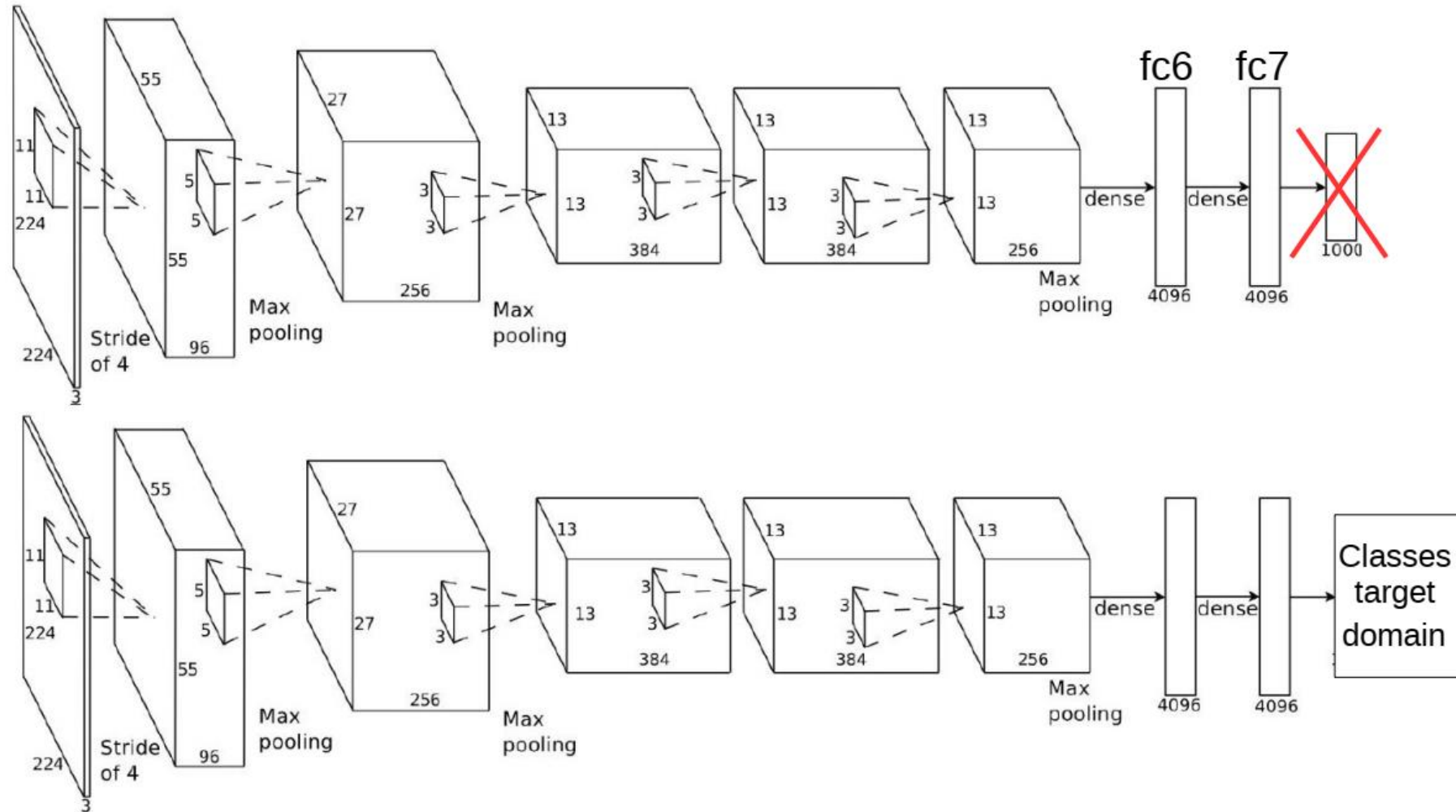


## Transferencia directa de features

Ex. A. Razavian et al., “*CNN Features off-the-shelf: an Astounding Baseline for Recognition*”, 2014.

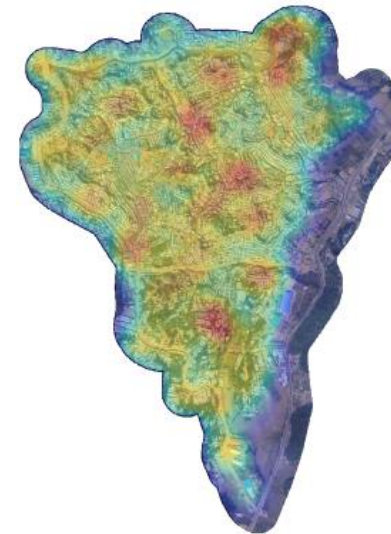
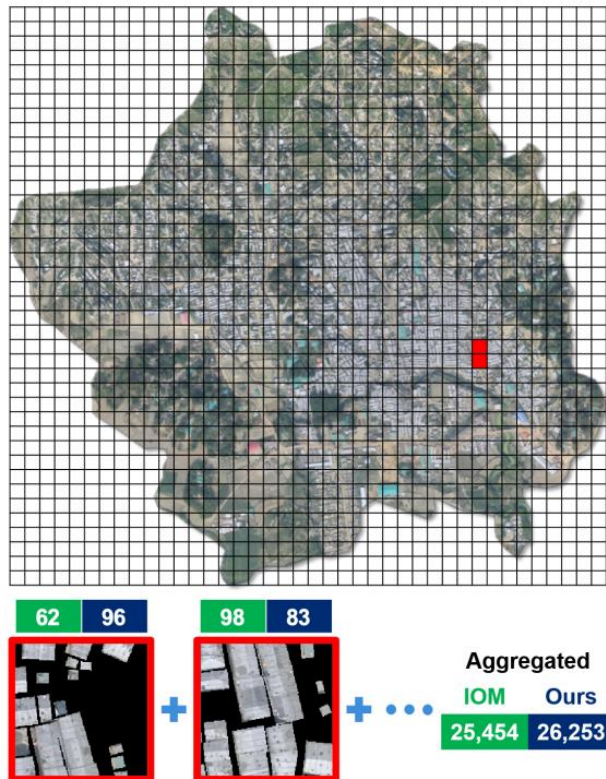


## Finetuning de features

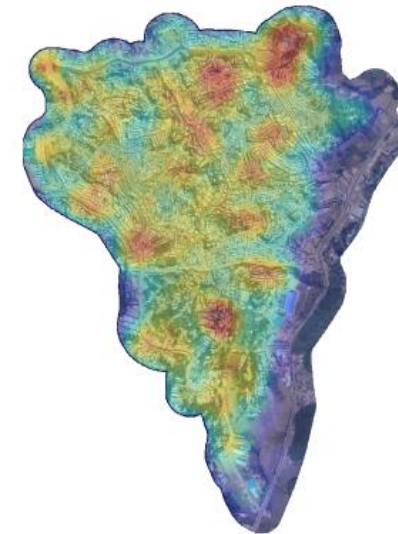


## Finetuning de features

**Estimating Displaced Populations From Overhead:** usando una CNN entrenada en ImageNet como base (ResNet-50), se cambia clasificador final por regresor y se reentrena usando imágenes satelitales.



(a) Our Prediction



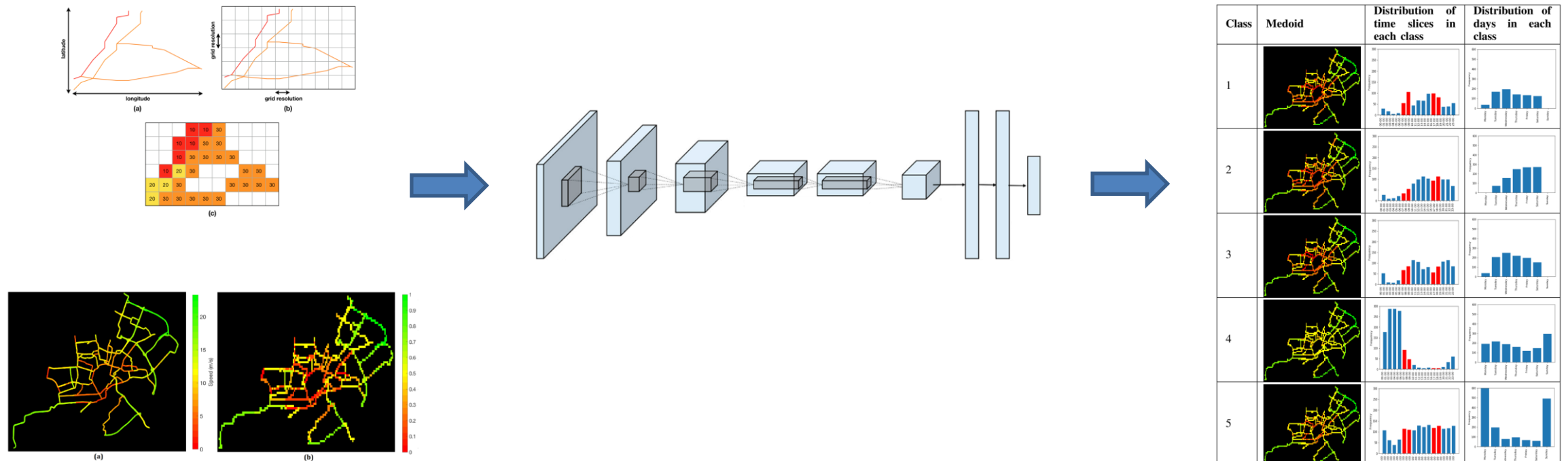
(b) IOM Estimate

## Finetuning presenta una gran cantidad de desafíos

- ¿Reentrenamos todas las capas o algunas? ¿Cuáles?
- ¿Cuándo decidir si se hace finetuning o entrenamiento desde cero de una red?
- ¿Cuán similares son los dominios?
- ¿Qué rol juega el *learning rate*? ¿Debe ser mayor o menor? ¿Es igual para todas las capas?

Para finalizar, ¿puedo combinar **ambos insights**?

**Understanding Network Traffic States using Transfer Learning:** se usa CNN entrenada en ImageNet como base (Inception-ResNet-v2) para extraer features de estados de tráfico codificados como imágenes y luego agruparlos.





Pontificia Universidad Católica de Chile  
Escuela de Ingeniería  
Departamento de Ingeniería de Transporte y Logística



# Sistemas Urbanos Inteligentes

Aspectos prácticos de CNNs

Hans Löbel

Dpto. Ingeniería de Transporte y Logística  
Dpto. Ciencia de la Computación