4. Redes Convolucionales

Abril 2024

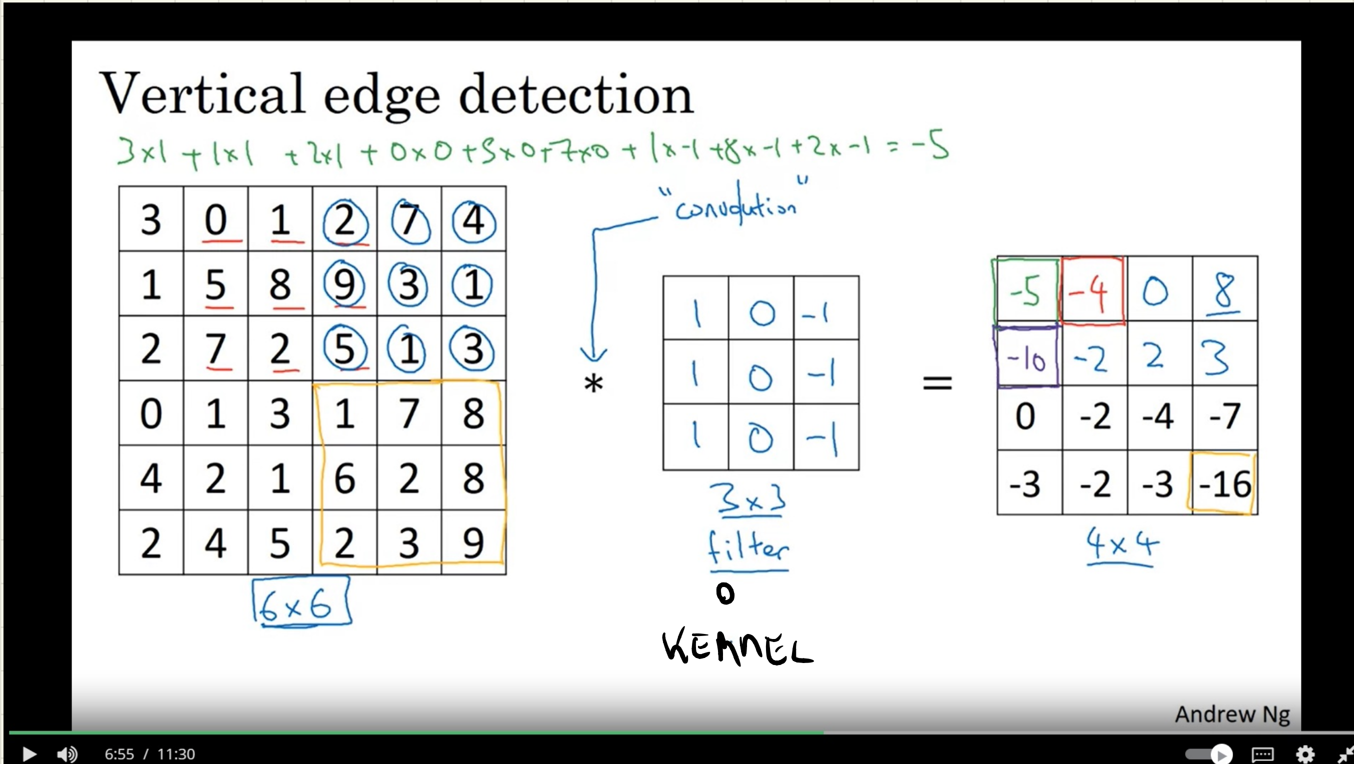
1 INTRODUCCIÓN

- Objetivos: clasificación de imágenes, detección de objetos, transferencia de estilos (combinar dos estilos.

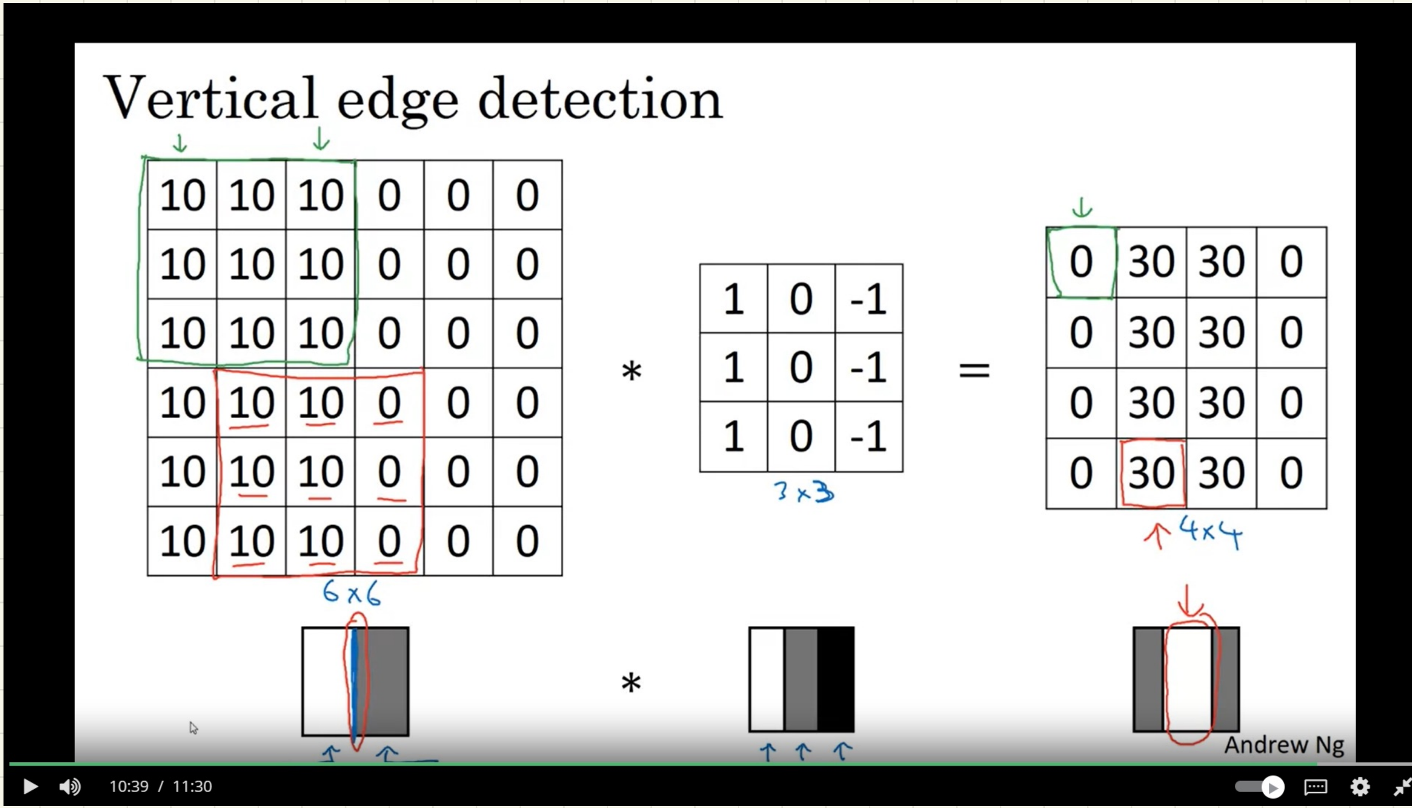
- Retos cada objeto puede ser muy grande.

- Una estrategia consiste en considerar las imágenes como una composición de imágenes más simple: bordes verticales, horizontales, etc

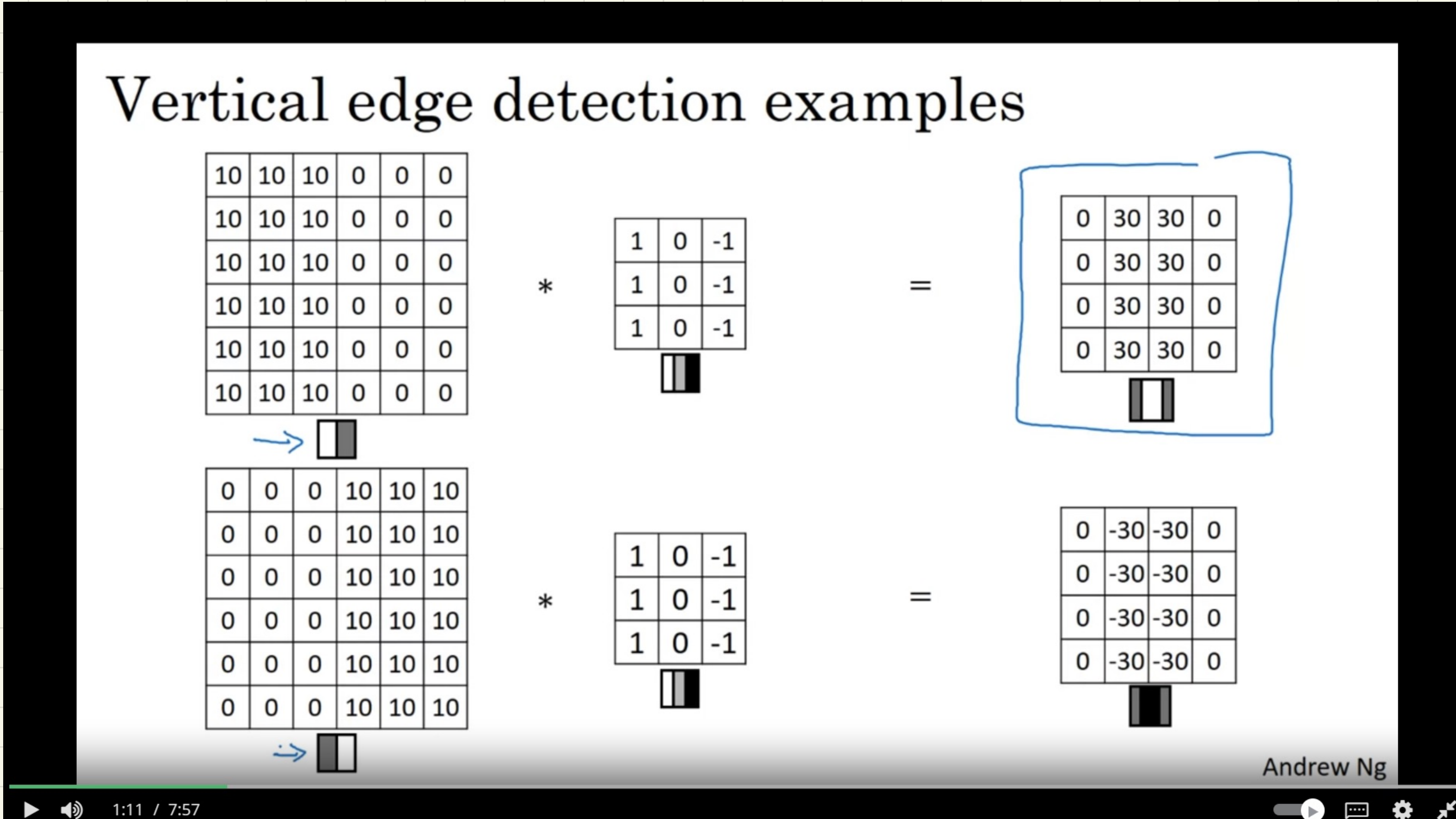
① convoluciones



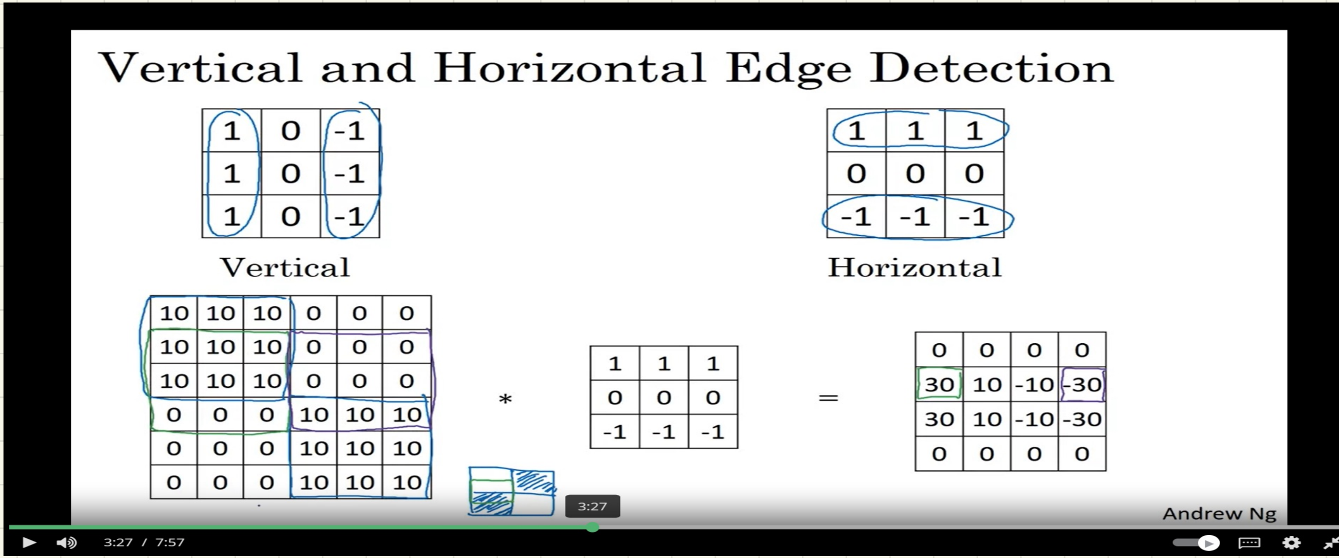
- Este filtro en particular ayuda a detectar bordes verticales:



- DIFERENCIAS DE PASAR DE GRIS A BLANCO - BLANCO A GRIS



- CASOS MÁS COMPLEJOS

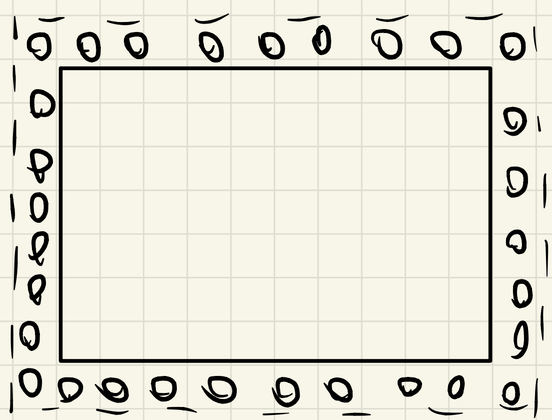


- EXISTEN OTRO TIPO DE FILTROS PARA DETECTAR BORDES VERTICALES / HORIZONTALES (VISIÓN POR COMPUTADOR)
- LA APROXIMACIÓN DE DL es aprender esos filtros

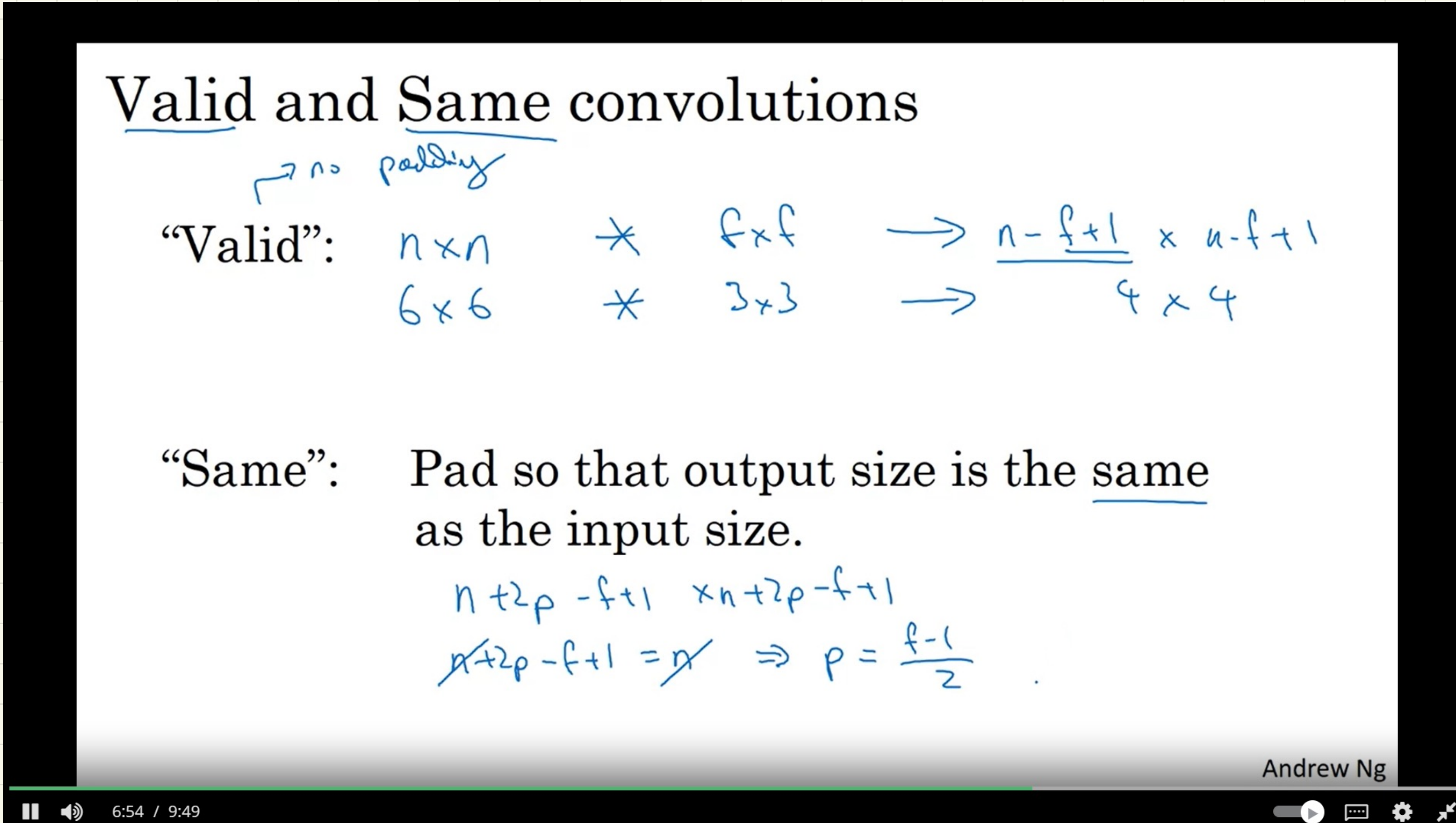
② PADDING

- Varios filtros pueden reducir el tamaño de la imagen
- Se pierde información en los bordes
- Solución:

padding = p = 1



- Valid and same convolutions



③ STRIDED CONVOLUTIONS

Strided convolution

2	3	7	4	6	2	9
6	6	9	8	7	4	3
3^{3}	4^{4}	8^{4}	3	8	9	7
7^{1}	8^{0}	3^{2}	6	6	3	4
4^{-1}	2^{0}	1^{3}	8	3	4	6
3	2	4	1	9	8	3
0	1	3	9	2	1	4

7×7

*

3	4	4
1	0	2
-1	0	3

3×3

stride = 2

=

91	100	83

1:17 / 8:57

Andrew Ng

Summary of convolutions

$n \times n$ image $\quad f \times f$ filter

padding $p \quad$ stride s

$$\left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor \times \left\lfloor \frac{n+2p-f}{s} + 1 \right\rfloor$$

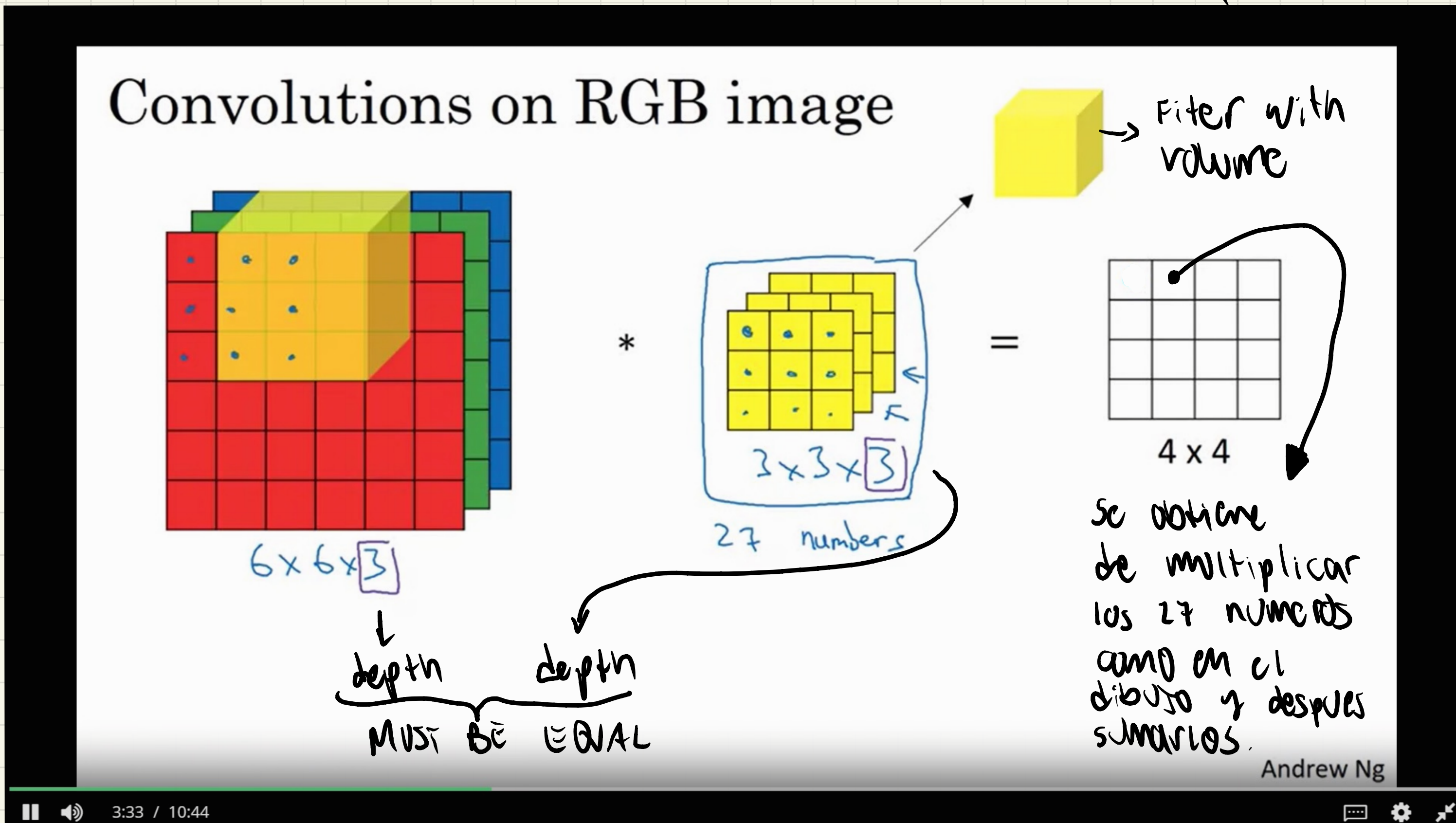
Andrew Ng

4:14 / 8:57

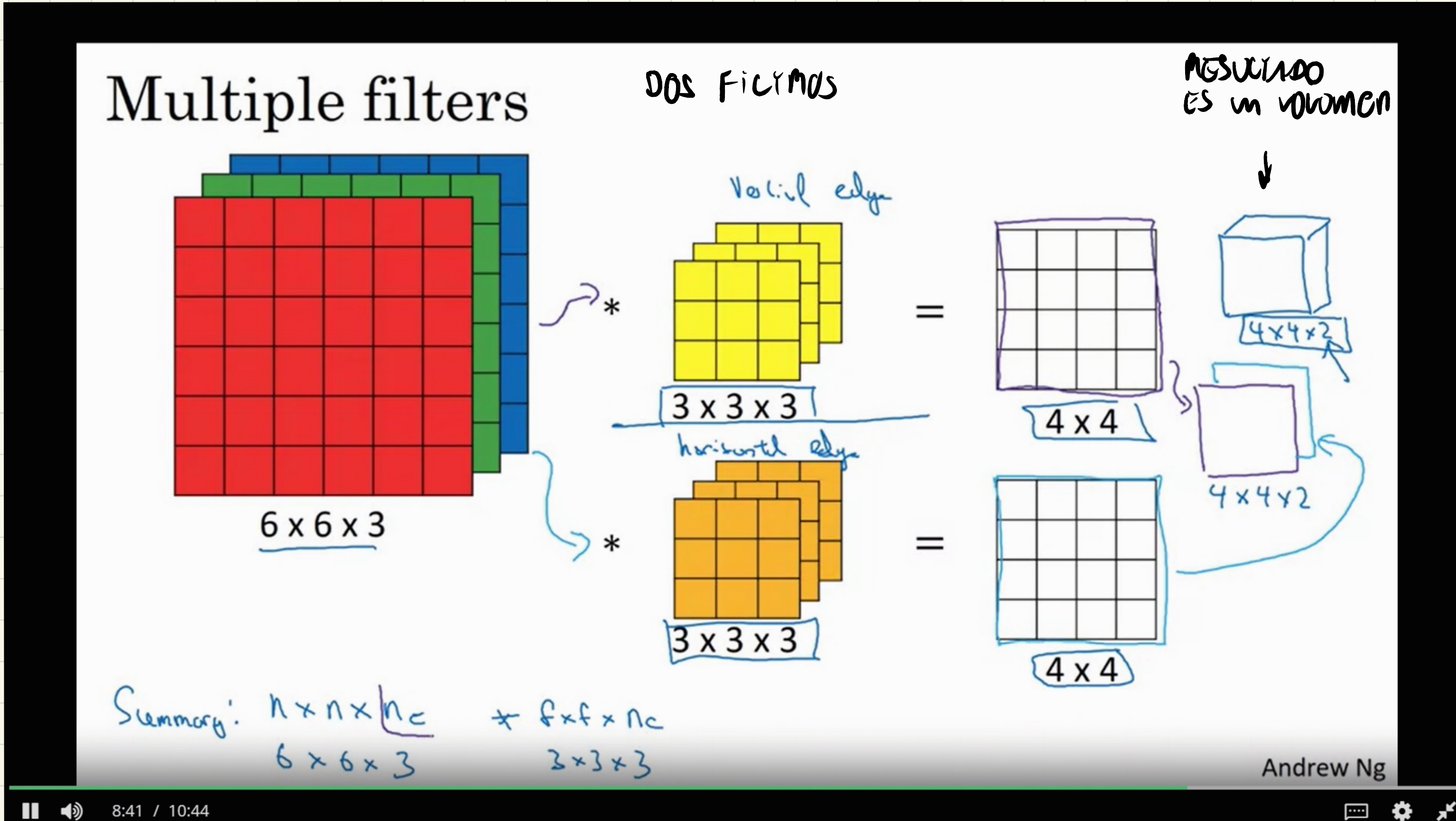
- EN MATEMATICAS ESTA OPERACION ES MAS PRECISAMENTE LLAMADA CORRELACION CRUZADA. CONVOLUCION ES LIGERAMENTE DISTINTO

④ Convolución de imágenes con múltiples canales

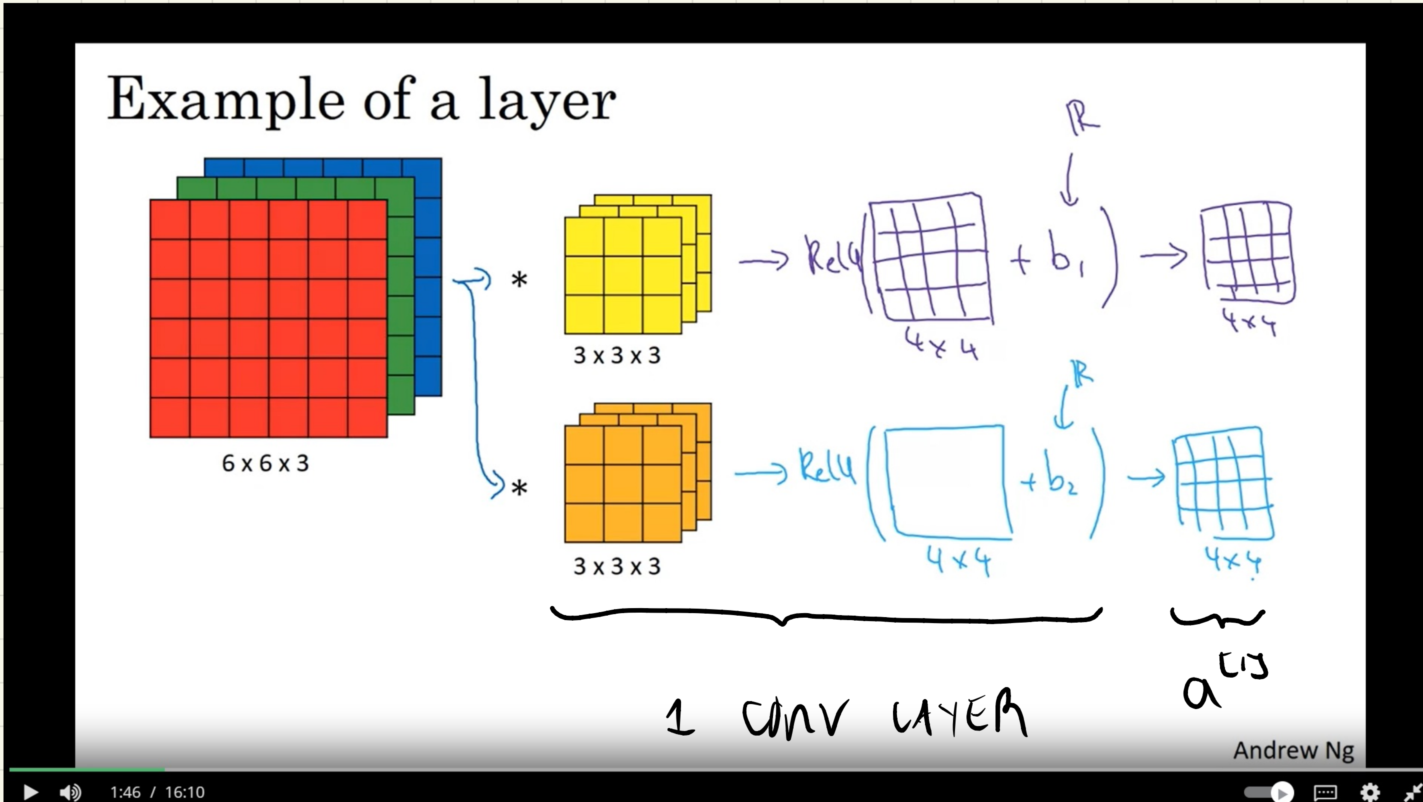
volumen



5 Convolucion con multiples filtros

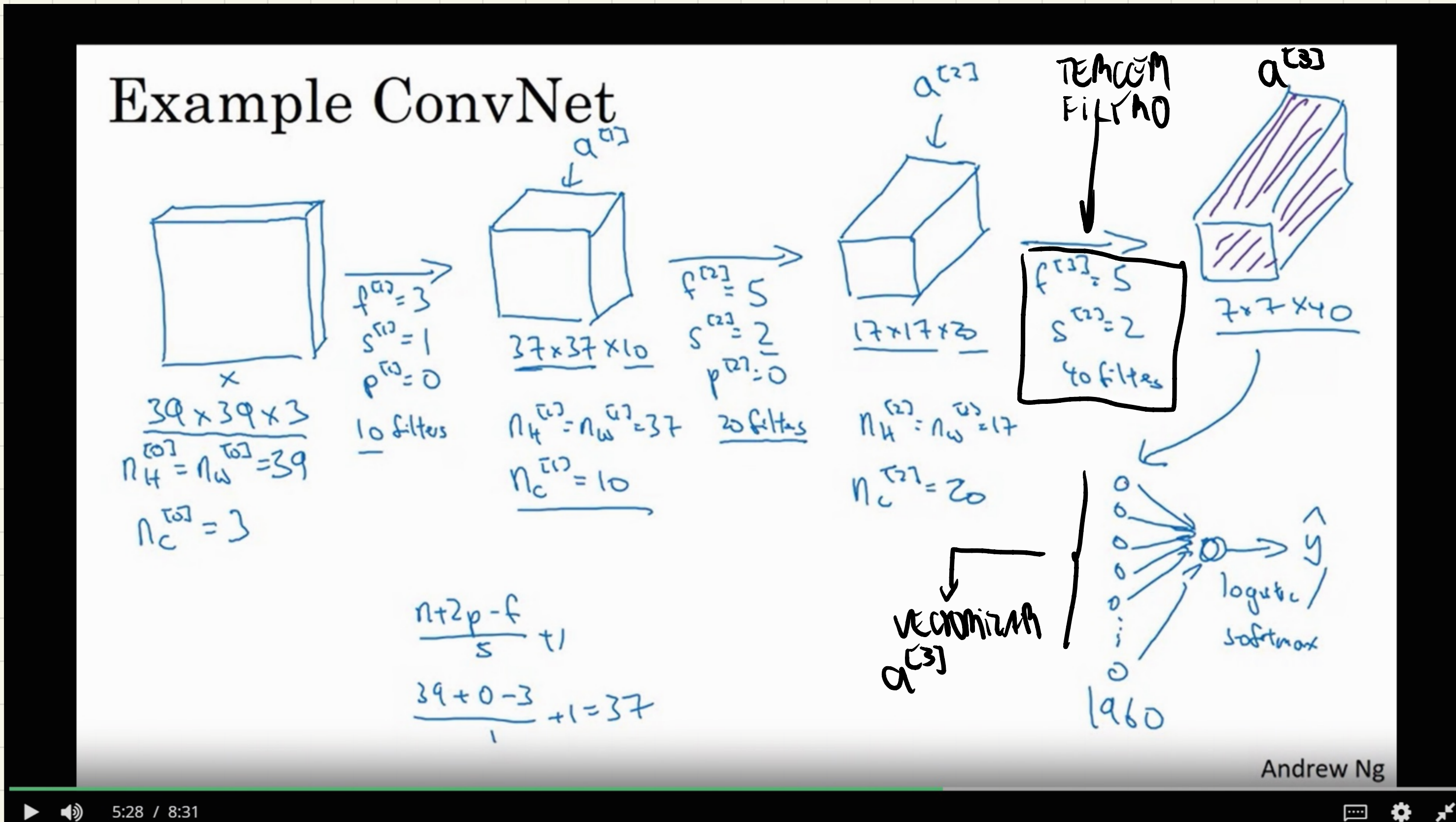


① RED NEURONAL CON UNA CAPA CONVOLUCIONAL



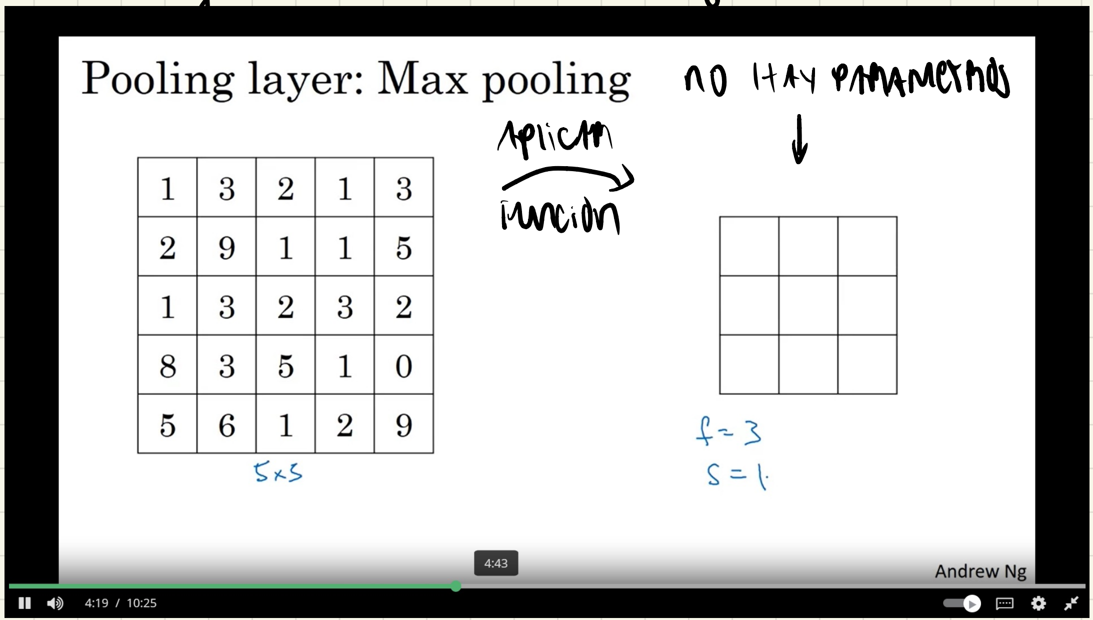
- observese que el número de parámetros de una capa no depende del tamaño de la imagen. solo de el número de filtros, su altura, ancho y profundidad + los sesgos que son tantos como filtros existen.

② EJEMPLO DE RED NEURONAL CONV PROFUNDA (PURA)



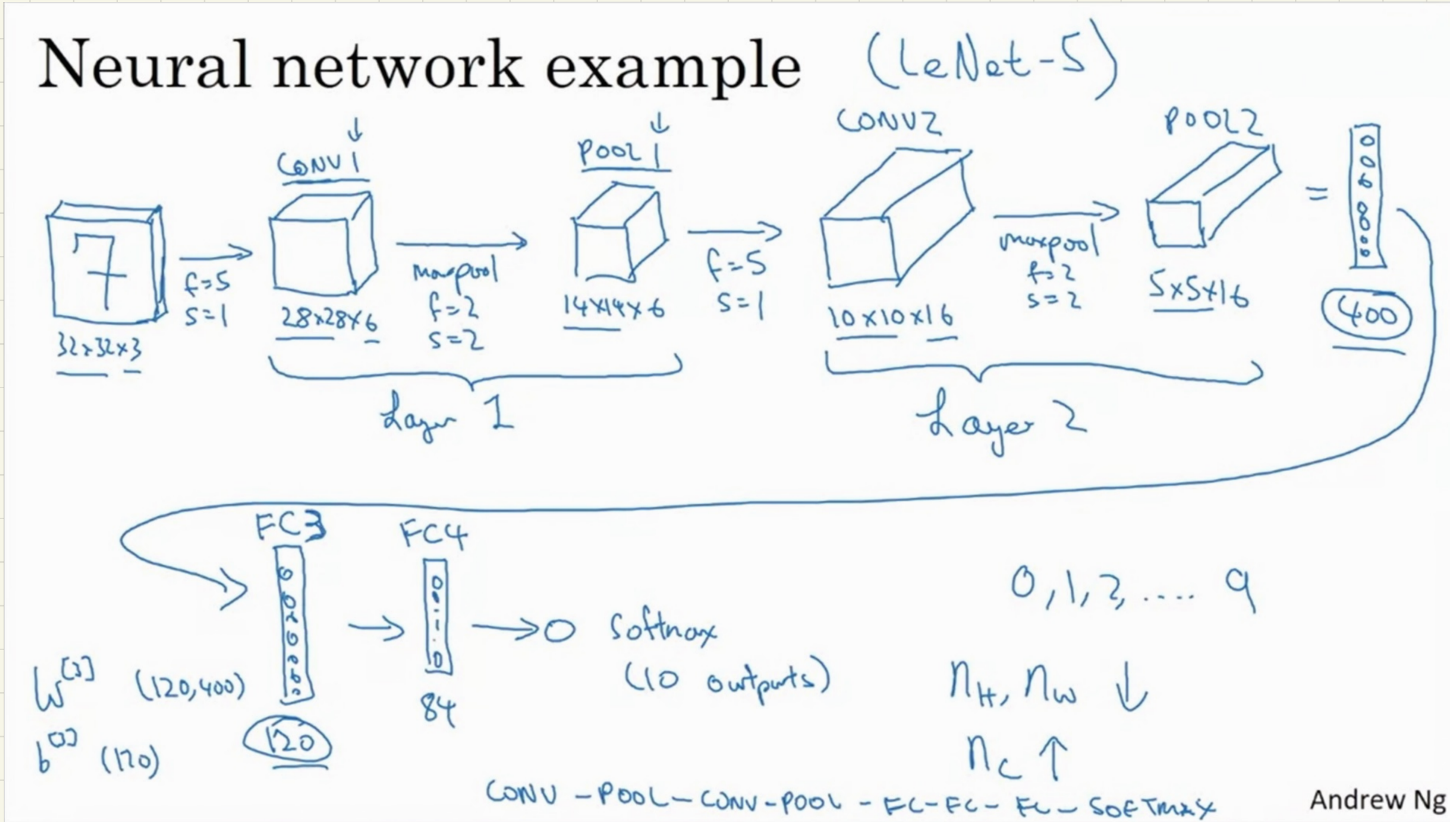
③ CAPA DE POOLING (HAS NO PARAMETERS TO LEARN)

- MAX POOLING / AVERAGE POOLING



- SE HACE FORMA INDEPENDIENTE PARA CADA CANAL.
- USUALMENTE NO SE USA PADDING

④ EJEMPLO DE RED NEURONAL CONV PROFUNDA



Ahora la convención es incluir la capa de pooling en la def. de capa convolucional

Neural network example

	Activation shape	Activation Size	# parameters
Input:	(32,32,3)	3,072 $a^{[0]}$	0
CONV1 (f=5, s=1)	(28,28,6)	4,704	456
POOL1	(14,14,6)	1,176	0
CONV2 (f=5, s=1)	(10,10,16)	1,600	2,416
POOL2	(5,5,16)	400	0
FC3	(120,1)	120	48,120
FC4	(84,1)	84	10,164
Softmax	(10,1)	10	850

5 ¿Por qué redes convolucionales?

- parámetros compartidos
- sparsity of connections: en cada capa sólo pocas inputs importan

① EJEMPLOS DE REDES

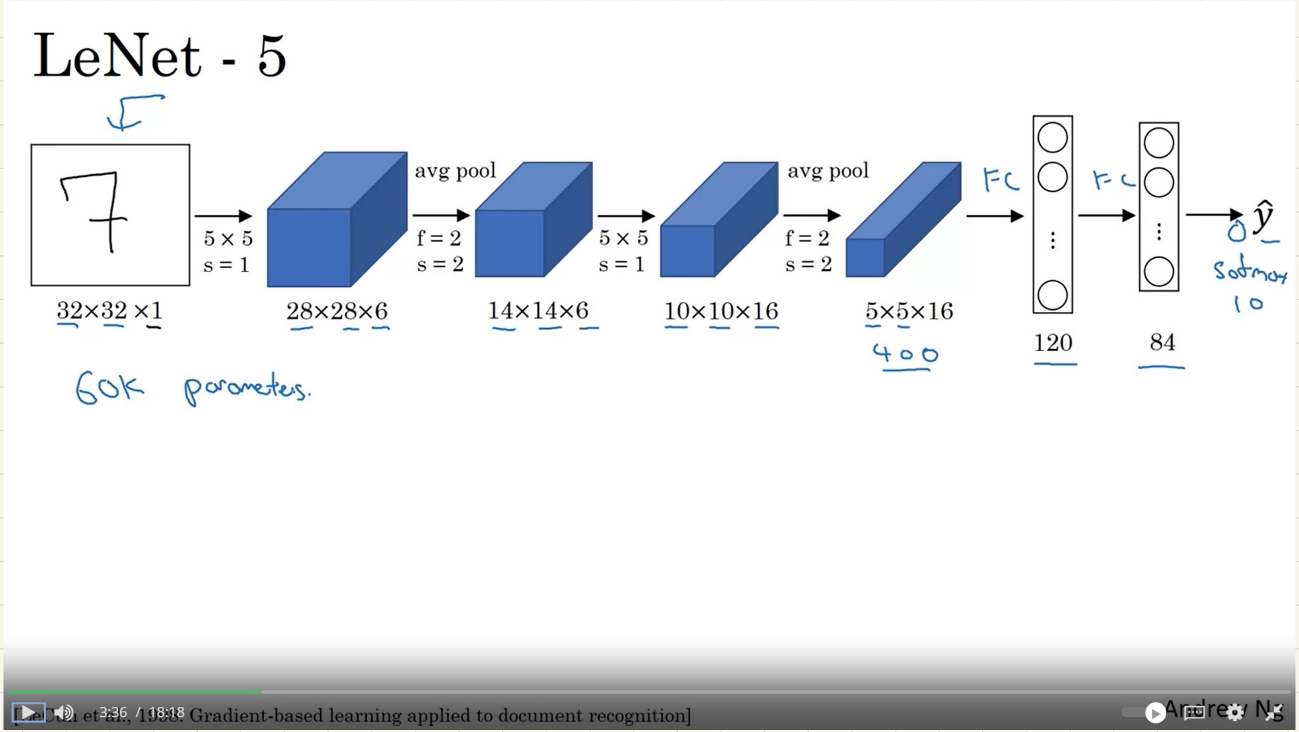
- LeNet-5
- AlexNet
- VGG

$\left.\begin{array}{l} \text{LeNet-5} \\ \text{AlexNet} \\ \text{VGG} \end{array}\right\}$ CLÁSICAS

- ResNet
- Inception

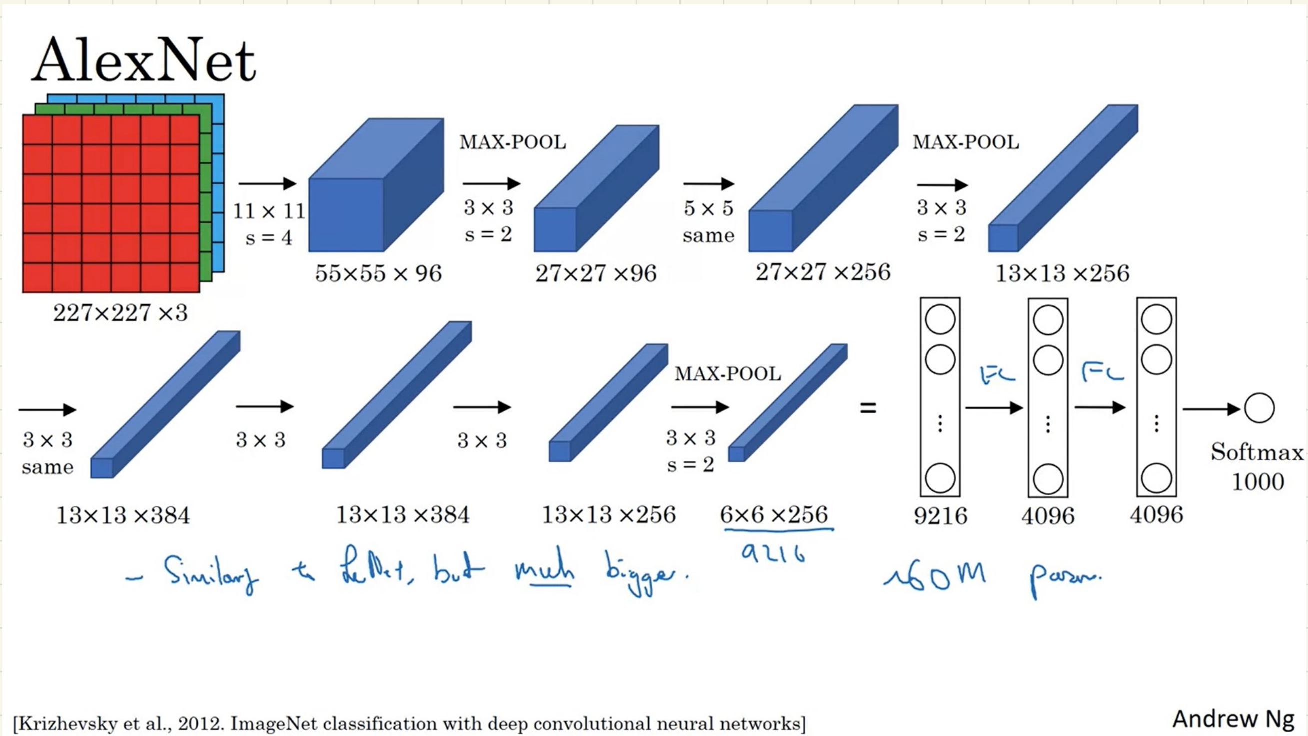
LeNet - 5   (LeCun 1988 : Gradient based learning applied to document recognition

- Históricamente una de las primeras redes en resolver exitosamente un problema: clasificación de dígitos.



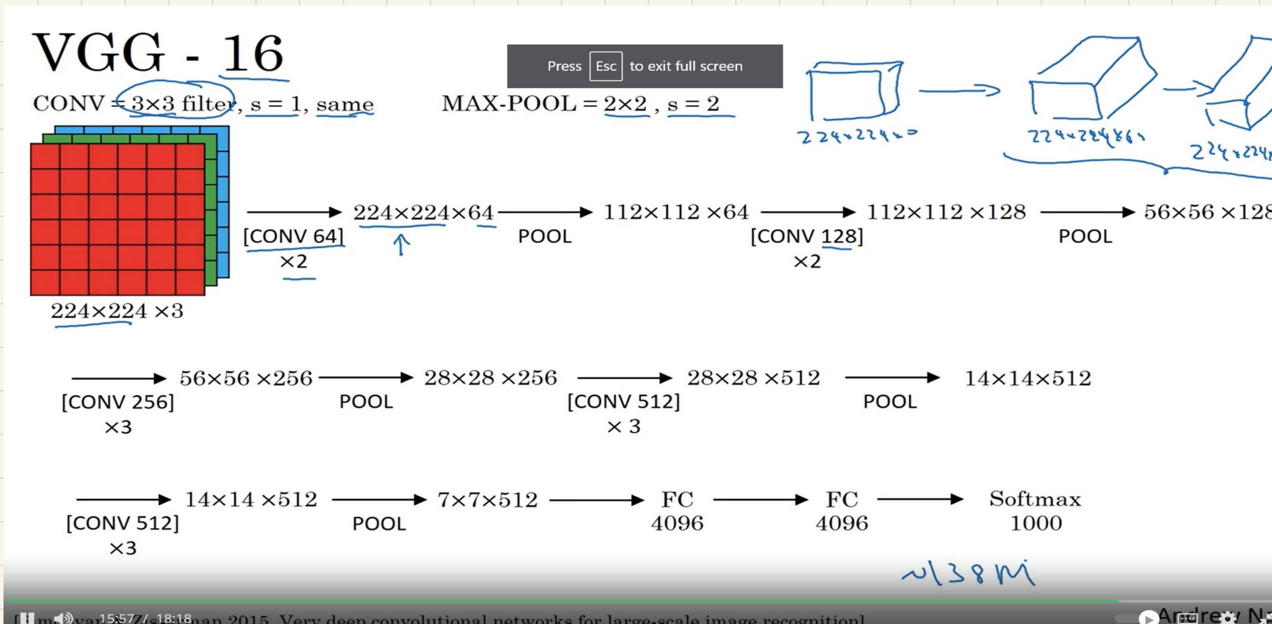
AlexNet : Krizhevsky 2012 : Image net classification with deep conv. neural network.

- Similar a LeNet-5 pero con 60 millones de parámetros.
- ESTE ARTICULO ES EL PUNTO DE INFLEXION EN CV USANDO DL



VGG-16 : Simonyan & Zisserman, 2015. Very deep conv. NN for large scale image recognition

- Simplifica las anteriores arquitecturas. Es más homogénea pero mucho más grande: 138 millones de parámetros.
- 16 capas con pesos



RESNET He 2015. Deep residual networks for IMAGE RECOGNITION

- PERMITE ENTRENAR REDES MUY GRANDES
- USA UN "SHORT CUT" O SKIP CONNECTION
- MITIGA EL PROBLEMA DE GRADIENTES $\to 0$ o $\to \infty$

DOS CAPAS DE UNA RED

$$a^{[l]} \underbrace{\xrightarrow{\text{linear}} \xrightarrow{\text{relu}} a^{[l+1]}}_{\text{CAPA 1}} \underbrace{\xrightarrow{\text{linear}} \xrightarrow{\text{relu}} a^{[l+2]}}_{\text{CAPA 2}}$$

$$a^{[l]} \xrightarrow{\text{linear}} \xrightarrow{\text{relu}} a^{[l+1]} \xrightarrow{\text{linear}} \xrightarrow{\text{relu}} a^{[l+2]}$$

$z^{[l+1]}$ $z^{[l+2]}$

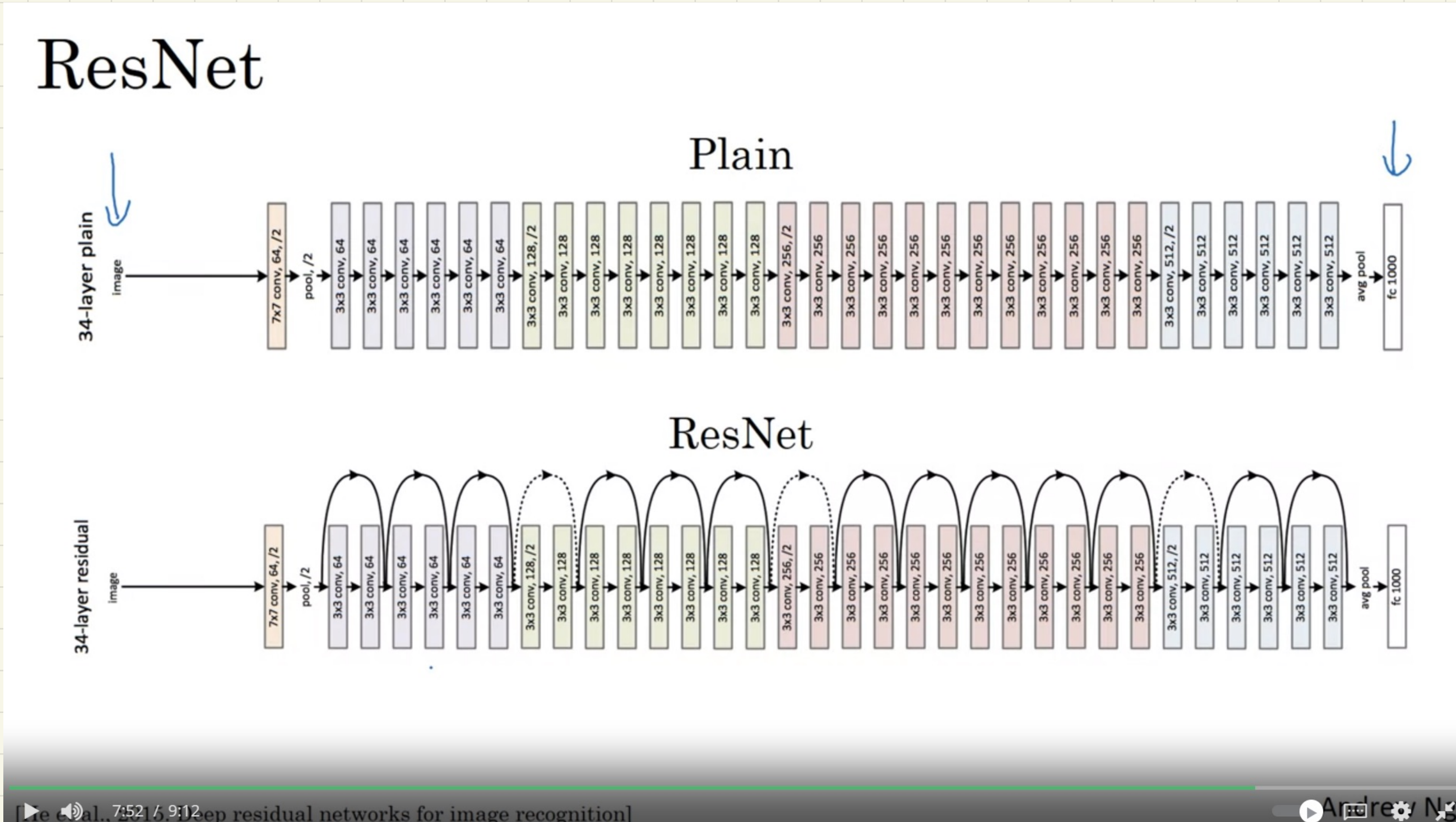
SHORT CUT

(RESIDUAL BLOCK)

$$a^{[l+2]} = g(z^{[l+2]} + a^{[l]})$$

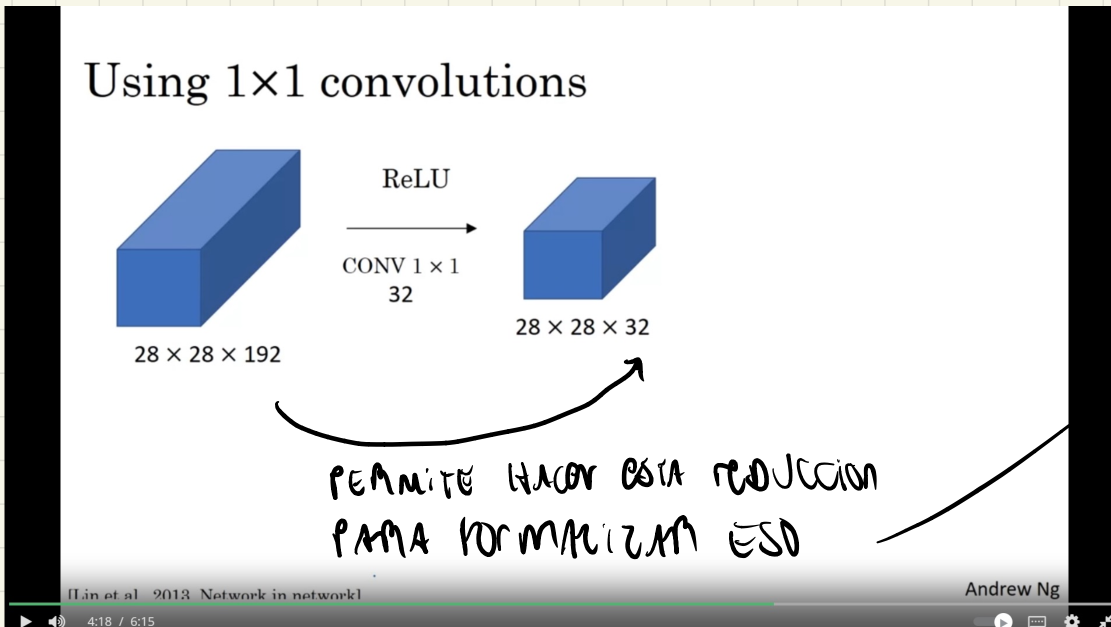
- HAY QUE TENER CUIDADO CON LAS DIMENSIONES PARA QUE

$a^{[l+k]} = g(z^{[l+k]} + a^{[l]})$ → HAGA SENTIDO

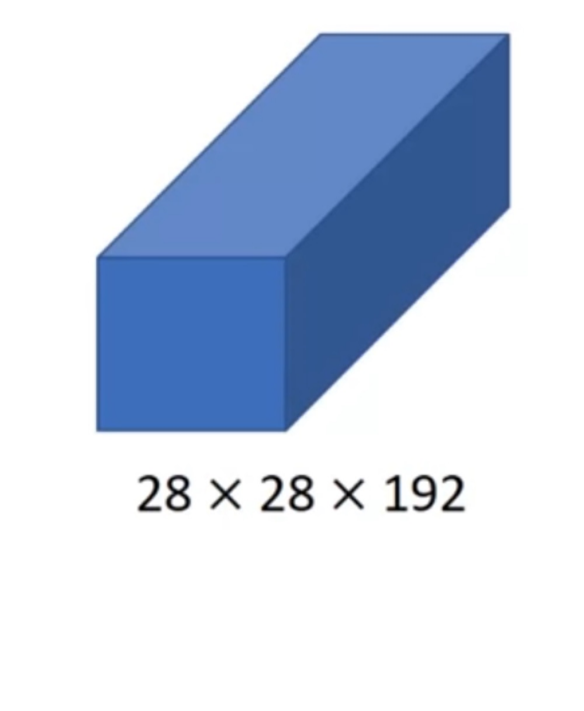


1x1 convoluciones

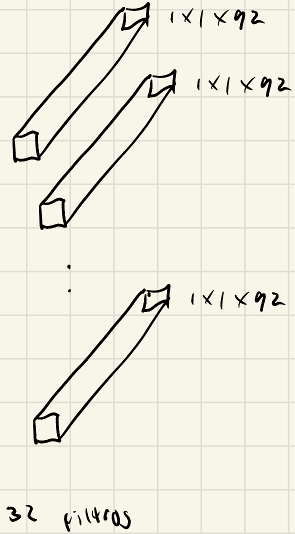
- Permite contraer o expandir el volumen
- Esto es relevante en Inception Net



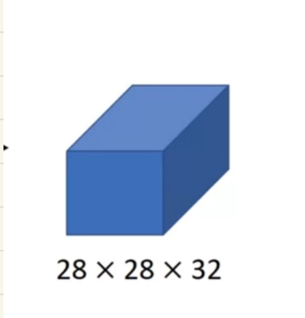
Permite hacer esta reducción para formalizar eso



*



Relu



Observación: En general no siempre se va reportar el volumen de los filtros porque tienen que ser igual al volumen del input.

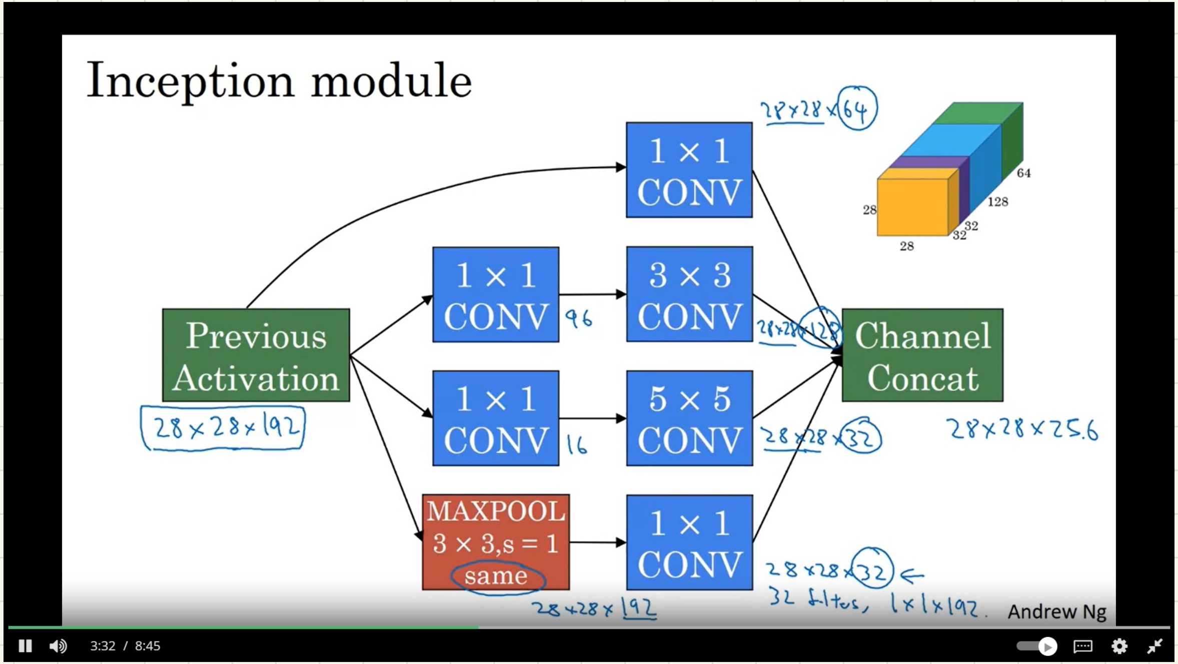
EN RESUMEN

- LAS CAPAS DE pooling se pueden usar para Agrandar o reducir Altura y ancho.

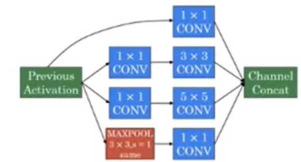
- Los filtros 1x1 para modificar el volumen.

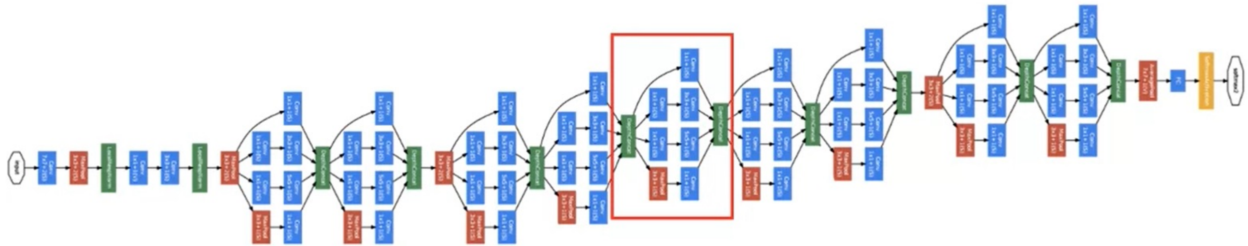
Inception Network : Szegedy et. al 2014 Going deeper in conv. networks.

- No es necesario decidir el tamaño de los filtros o capas de pooling. se ponen todas.



Inception network





[Szegedy et al., 2014, Going Deeper with Convolutions]

Andrew Ng

4:02 / 8:45

- <u>Mobile net</u>

ES UNA ARQUITECTURA QUE PUEDE DESPLEGARSE CON POCA CAPACIDAD DE COMPUTO.

ESTA BASADA EN UNA FORMA MAS EFICIENTE DE HACER LOS CALCULOS DE CONVOLUCION <u>separando en dos etapas</u>

① "DEPTHWISE" CONVOLUTION

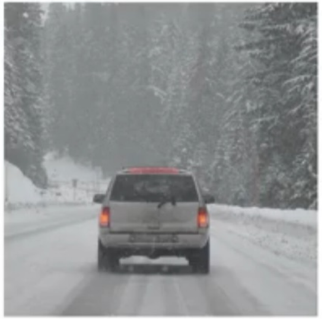
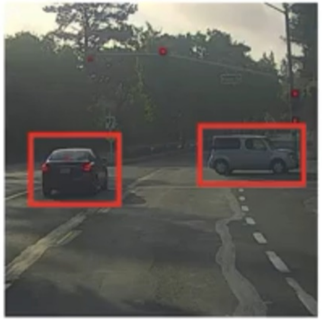
② "POINTWISE" CONVOLUTION

- <u>Efficient net</u>

ES UNA ARQUITECTURA QUE PERMITE ELEGIR ARQUITECTURAS DE MENOS A MAS RECURSOS COMPUTACIONALES.

TAREA : CLASIFICACIÓN , LOCALIZACIÓN , DETECCIÓN

What are localization and detection?

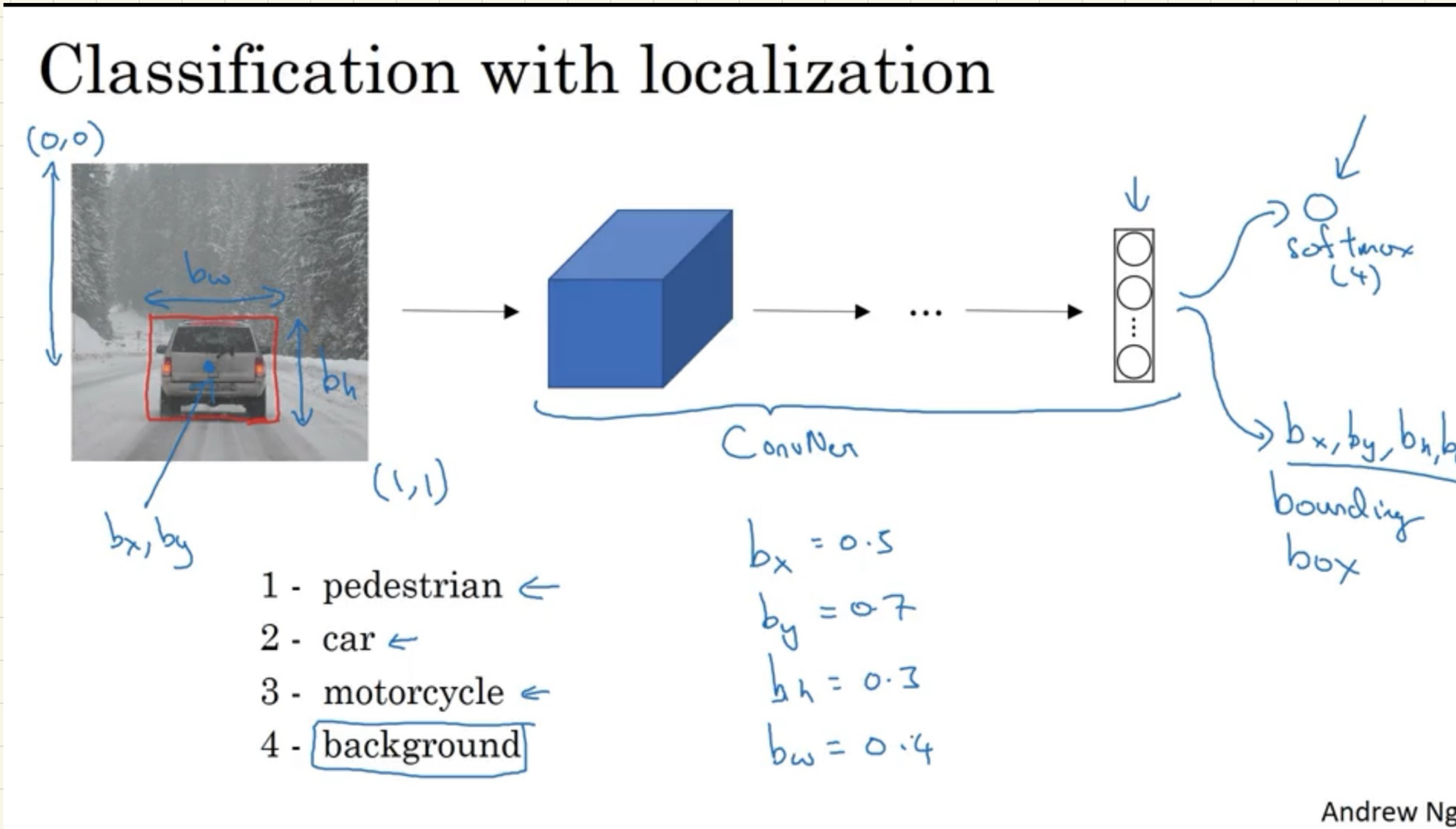
Image classification	Classification with localization	Detection
		
"Car"	"Car"	

1 object

multiple objects

Andrew Ng

TAREA : CLASIFICACIÓN, LOCALIZACIÓN



TAREA : CLASIFICACIÓN, LOCALIZACIÓN

Defining the target label y

1 - pedestrian
2 - car
3 - motorcycle
4 - background

Need to output b_x, b_y, b_h, b_w, class label (1-4)

$x =$

$$\mathcal{L}(\hat{y}, y) = \begin{cases} (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \dots + (\hat{y}_8 - y_8)^2 & \text{if } y_1 = 1 \\ (\hat{y}_1 - y_1)^2 & \text{if } y_1 = 0 \end{cases}$$

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

is there any object?

$$\begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

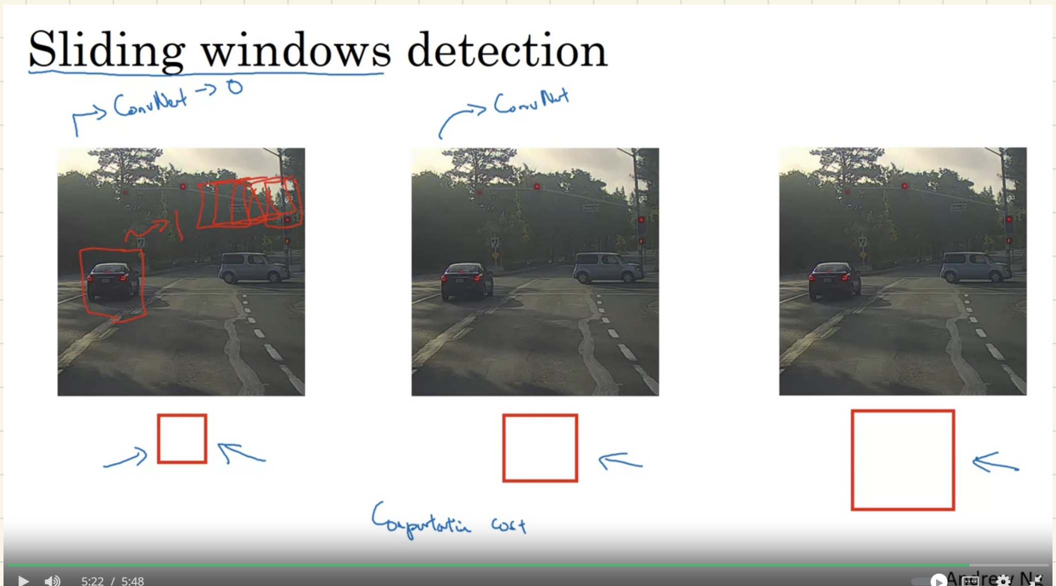
"don't care"

(x, y)

Andrew Ng

TAREA: DETECCION

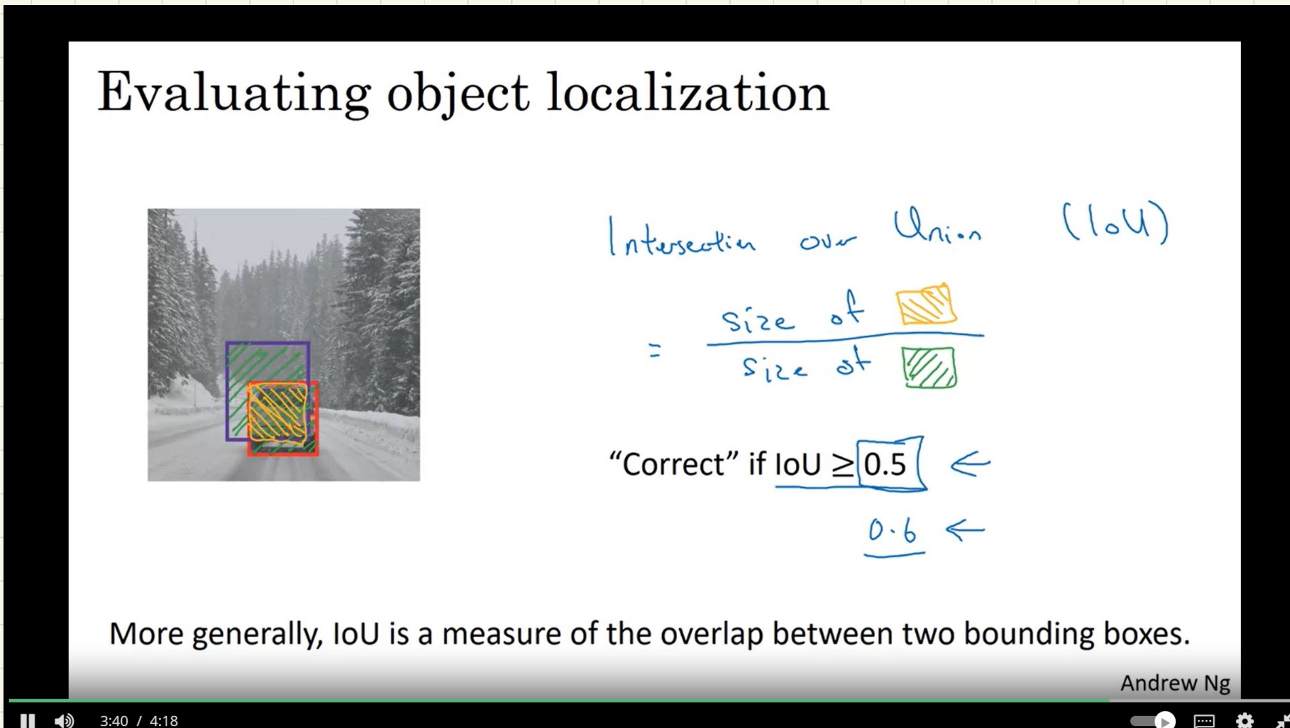
- ENTRENAR UN ALGORITMO DE CLASIFICACION
- USAR "SLIDING WINDOW DETECTION". Puede implementarse de forma convolucional: TODAS LAS PREDICCIONES se HACEN DE FORMA SIMULTÁNEA.



TAREA : **BOUNDING BOX PREDICTIONS**

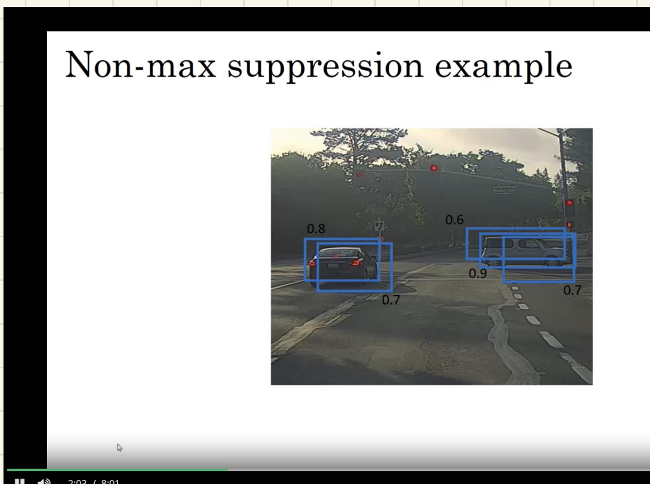
TAREA: EVALUACIÓN DE QUÉ TAN BUENA ES LA LOCALIZACIÓN DE UN OBJETO

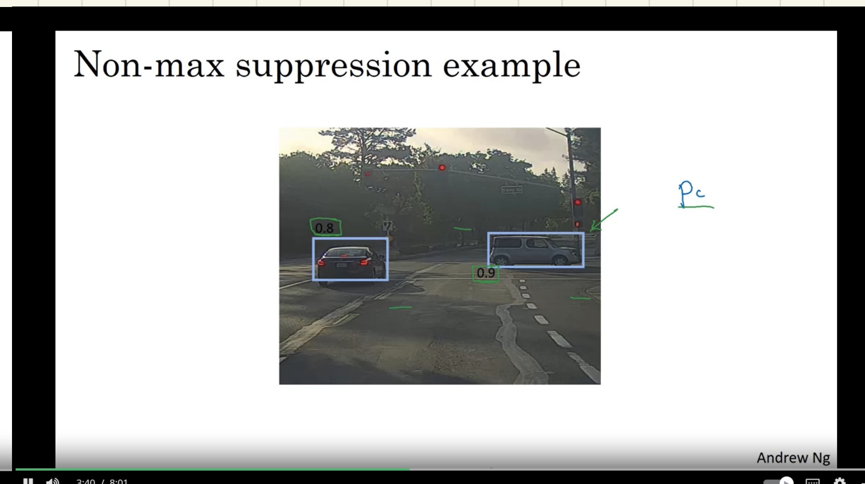
Métrica: INTERSECCIÓN SOBRE UNIÓN



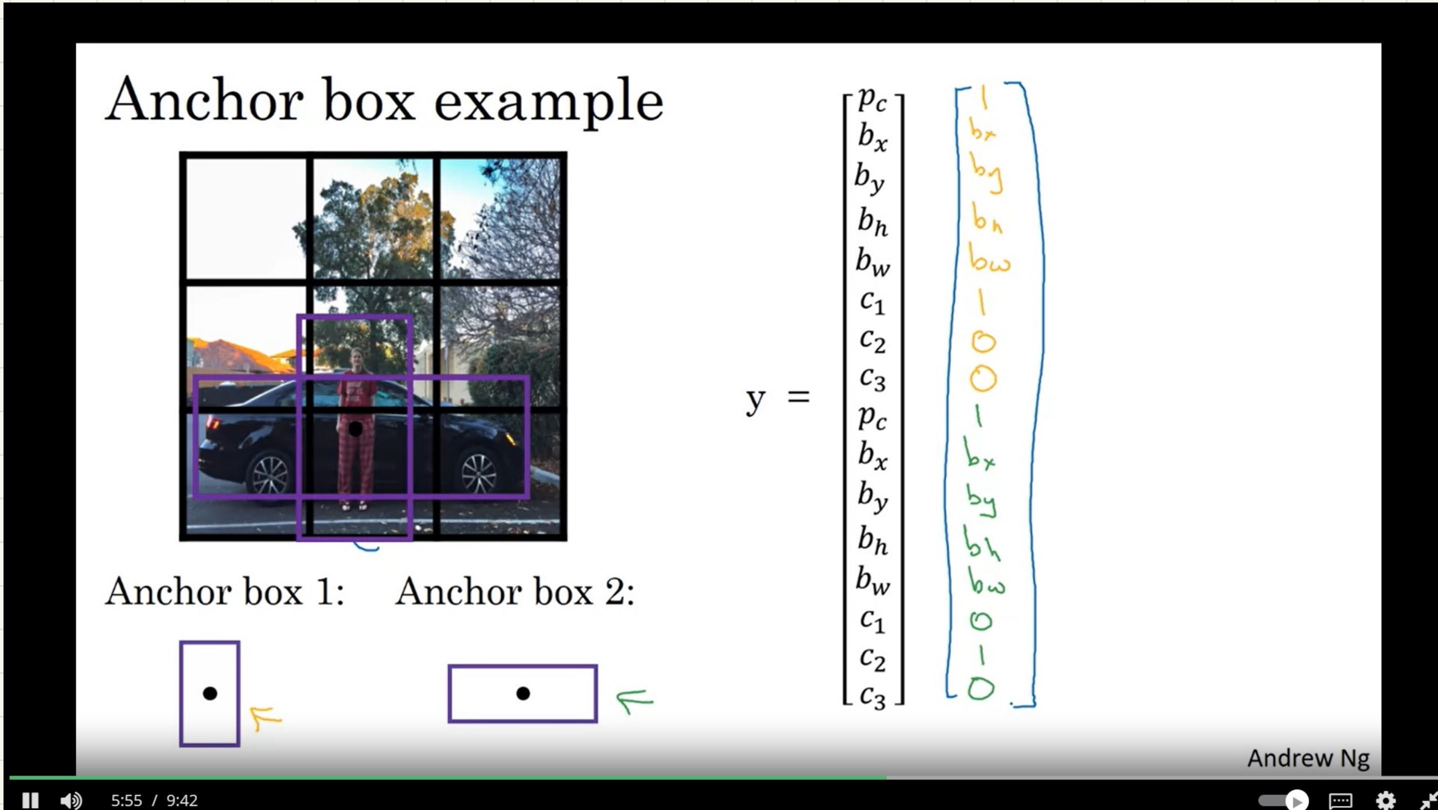
- Un problema con las técnicas hasta este punto es que se detecta varias veces un objeto.
- <u>Non max suppression</u> : consiste en :

1. Descartar todas las cajas con prob. baja ≤ 0.6
2. Elegir la que tenga prob. alta
3. Descartar todas las que tengan $IoU \geq 0.5$

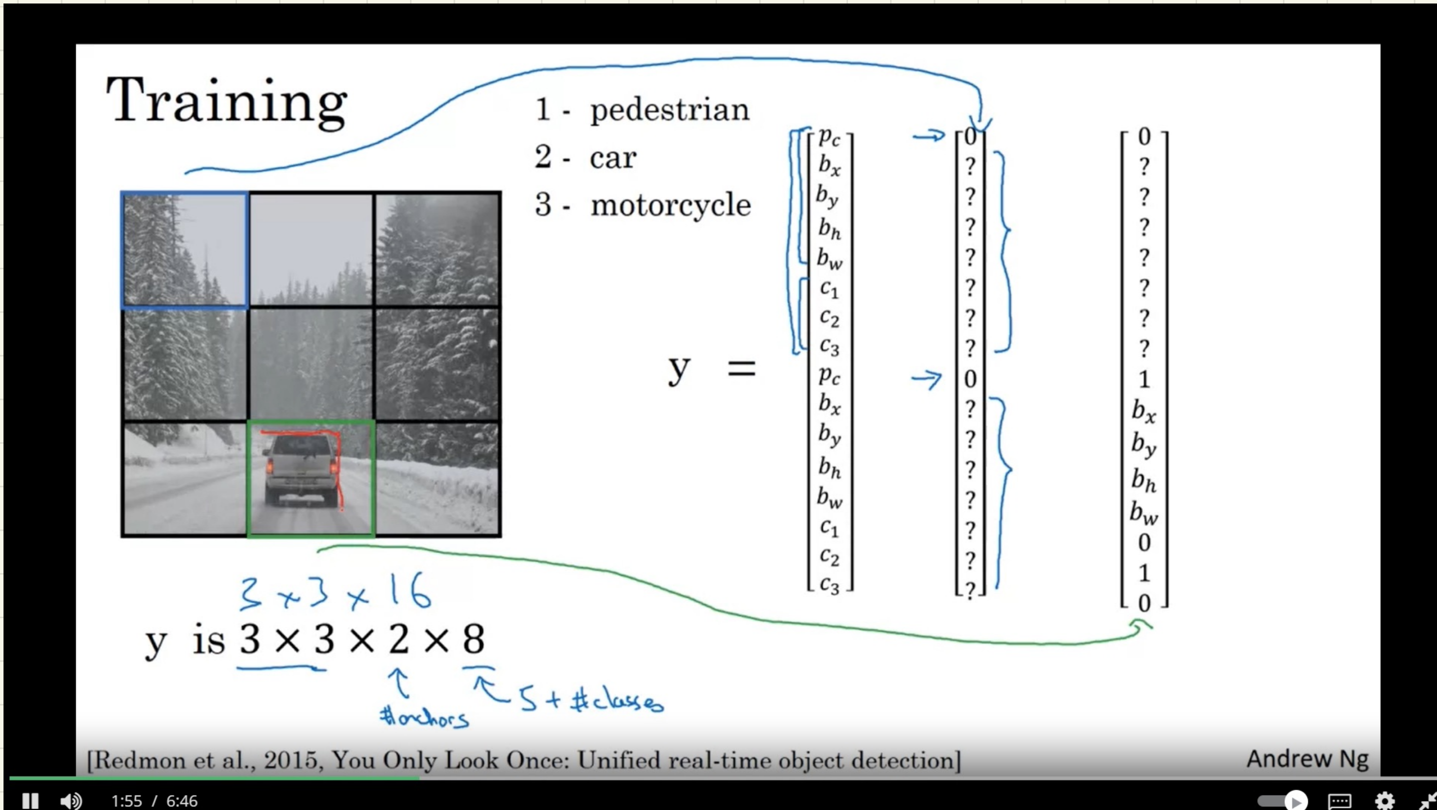




TAREA : DETECCION DE MULTIPLES OBJETOS

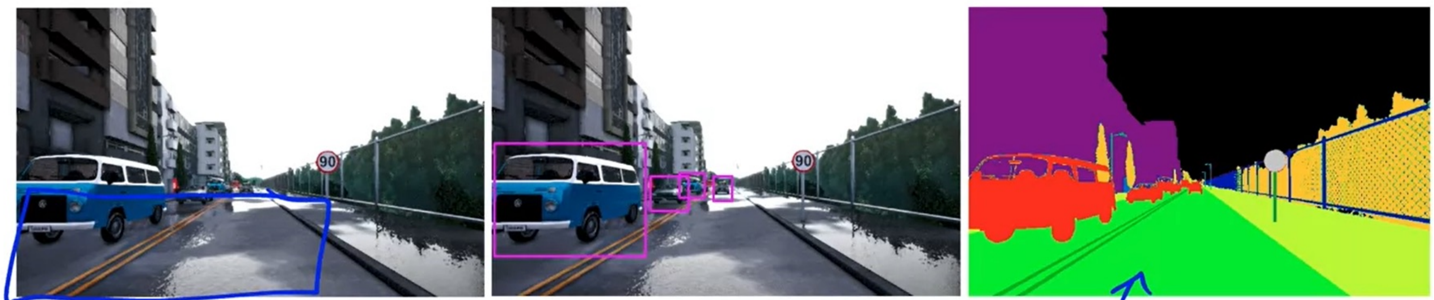


YOLO ALGORITHM PONE TODO TECNICAS EN UN ALGORITMO



SEMANTIC SEGMENTATION

Object Detection vs. Semantic Segmentation

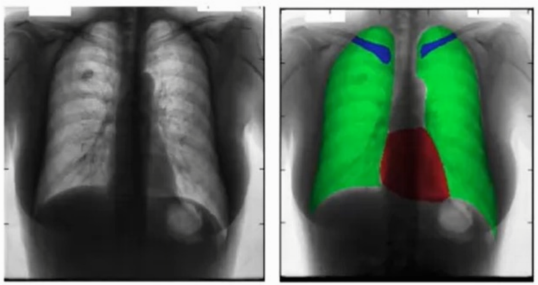


Input image | Object Detection | Semantic Segmentation

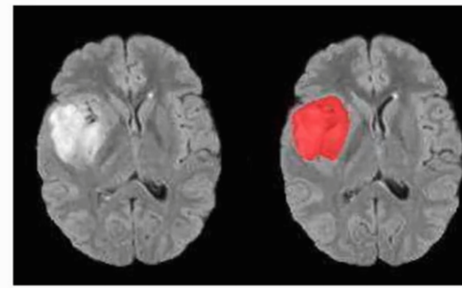
1:38 / 7:21

Andrew Ng

Motivation for U-Net



Chest X-Ray

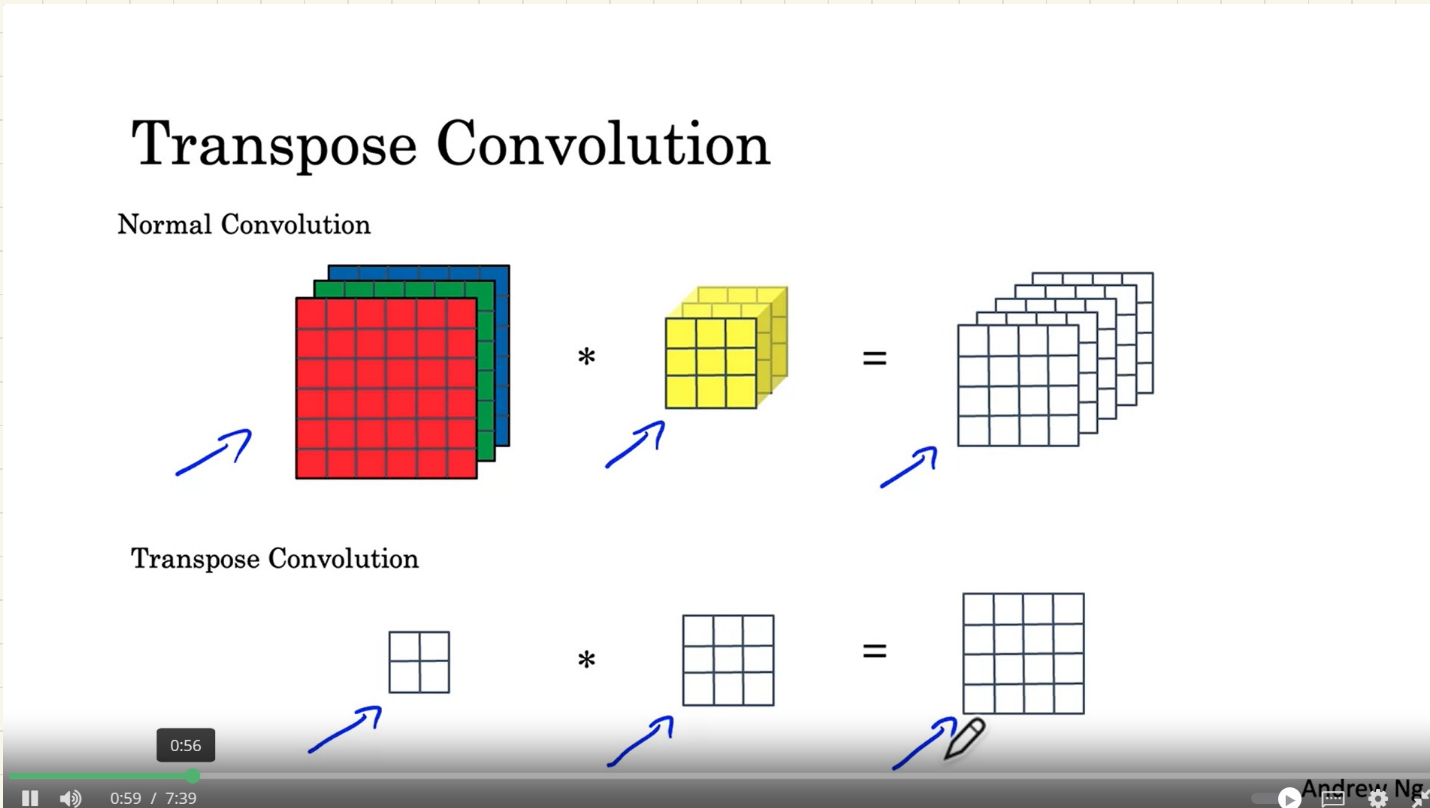


Brain MRI

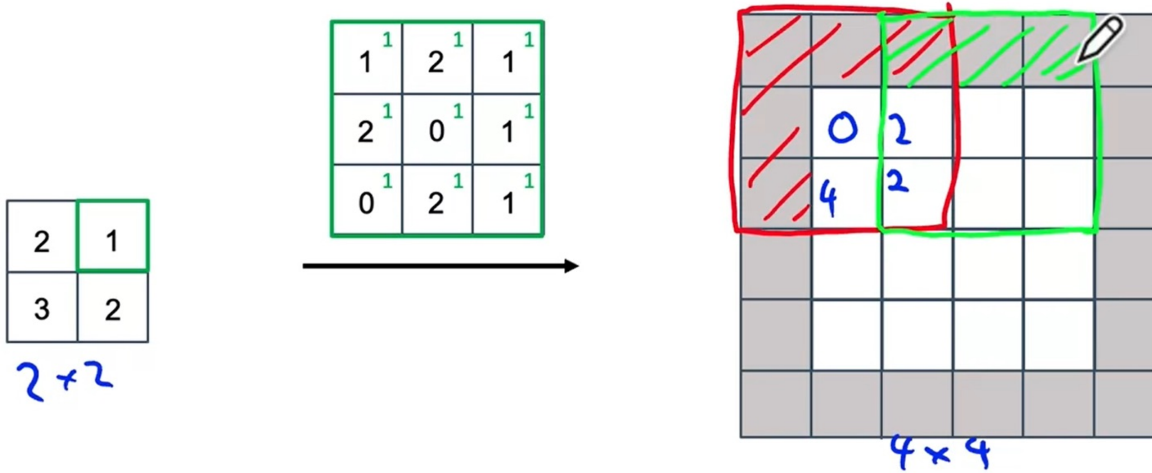
[Novikov et al., 2017, Fully Convolutional Architectures for Multi-Class Segmentation in Chest Radiographs]
[Dong et al., 2017, Automatic Brain Tumor Detection and Segmentation Using U-Net Based Fully Convolutional Networks]

Andrew Ng

TRANSPOSE CONVOLUTION



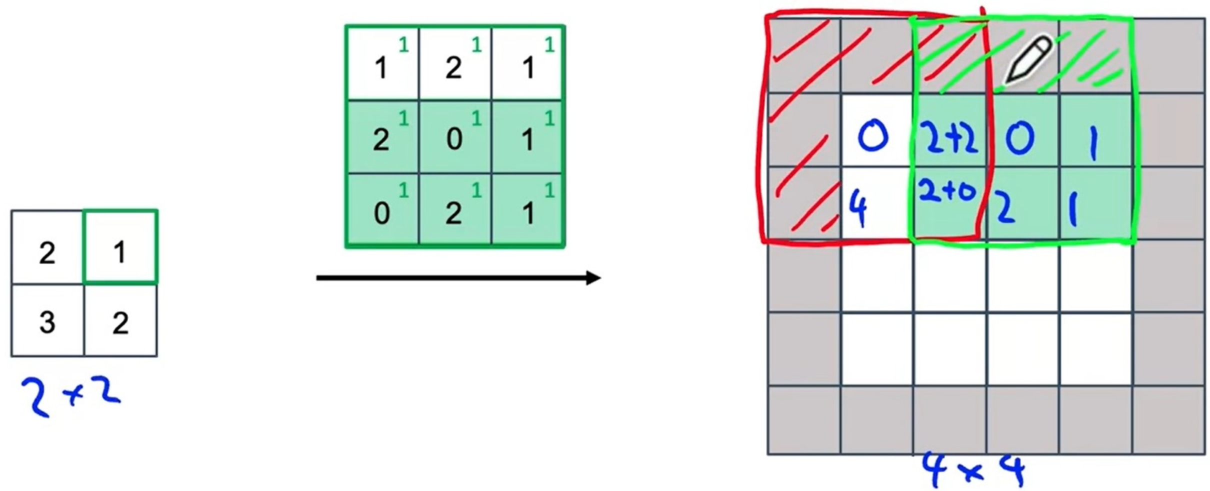
Transpose Convolution



filter $f \times f = 3 \times 3$ padding $p=1$ stride $s=2$

Andrew Ng

Transpose Convolution

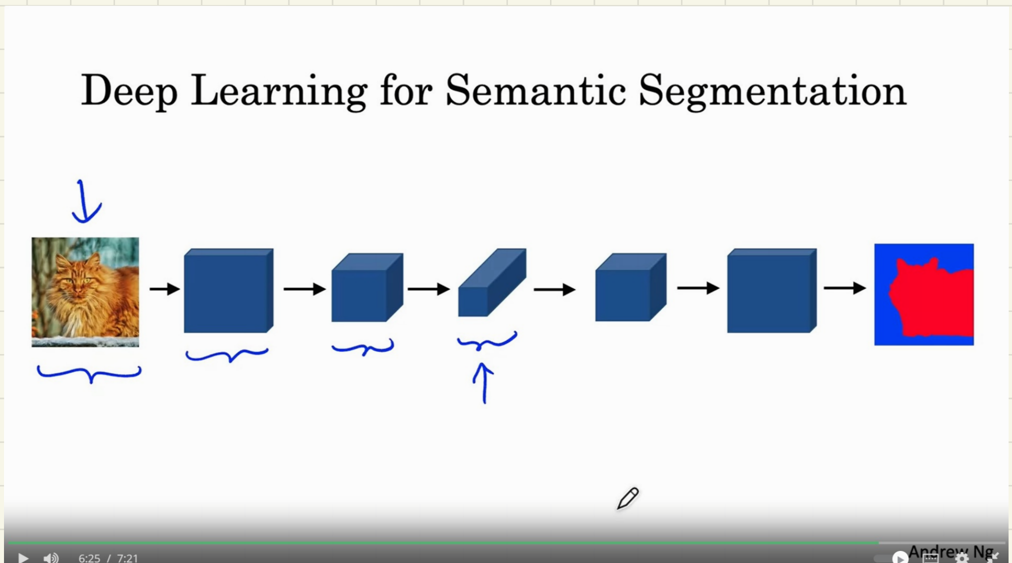


filter $f \times f = 3 \times 3$ padding $p = 1$ stride $s = 2$

Andrew Ng

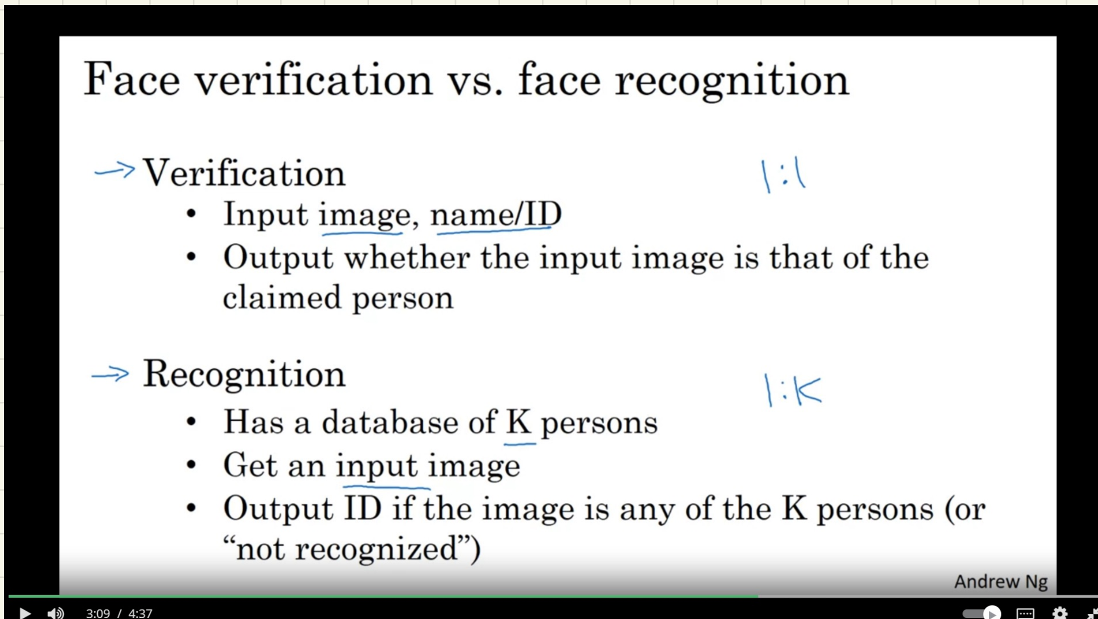
U - Net

- Esta arquitectura se apalanca en transpose convolution para resolver el problema de segmentación semántica.



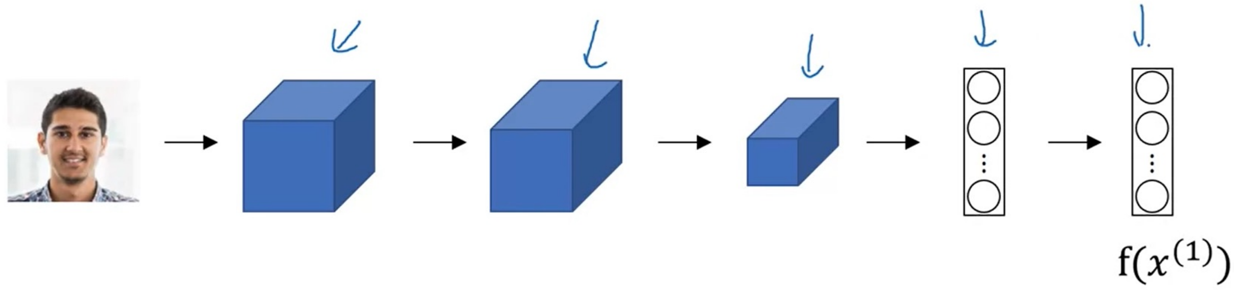
RECONOCIMIENTO DE CARAS : VERIFICACION vs reconocimiento

- One-shot learning: con un solo ejemplo reconocer si la persona esta en lista
- Estrategia aprender una funcion de similaridad: $d(img1, img2)$



- Siamese network

Goal of learning



Parameters of NN define an encoding $f(x^{(i)})$ — 128

Learn parameters so that:

If $x^{(i)}, x^{(j)}$ are the same person, $\left\| f(x^{(i)}) - f(x^{(j)}) \right\|^2$ is small.

If $x^{(i)}, x^{(j)}$ are different persons, $\left\| f(x^{(i)}) - f(x^{(j)}) \right\|^2$ is large.

- Funcion objetivo : TRIPLET LOSS FUNCTION

- Los datos se organizan en tripletes : (A, P, N)

Anchor

positive (SAME)

Negative (diferent)

Hay que tener cuidado porque si se hace aleatorio en facil minimizar la perdida.

- $\mathcal{L}(A, P, N)$

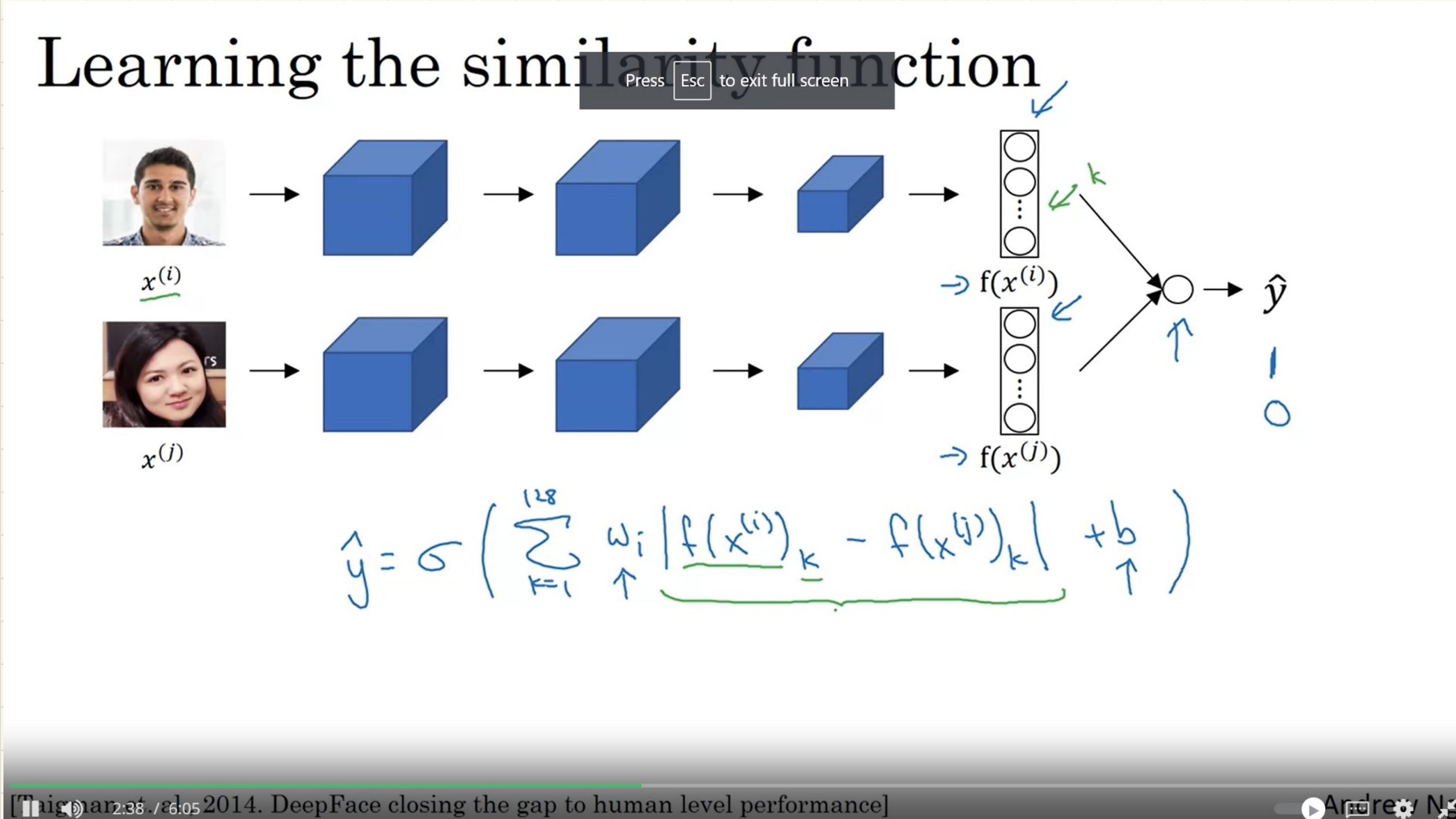
$$= \max \{ \|f(A) - f(P)\|^2 - \|f(A) - f(N)\|^2 + \alpha , 0 \}$$

n-encoding de siamese arquitecture

- $$J = \sum_{i=1}^{m} \mathcal{L}(A^{(i)}, P^{(i)}, N^{(i)})$$

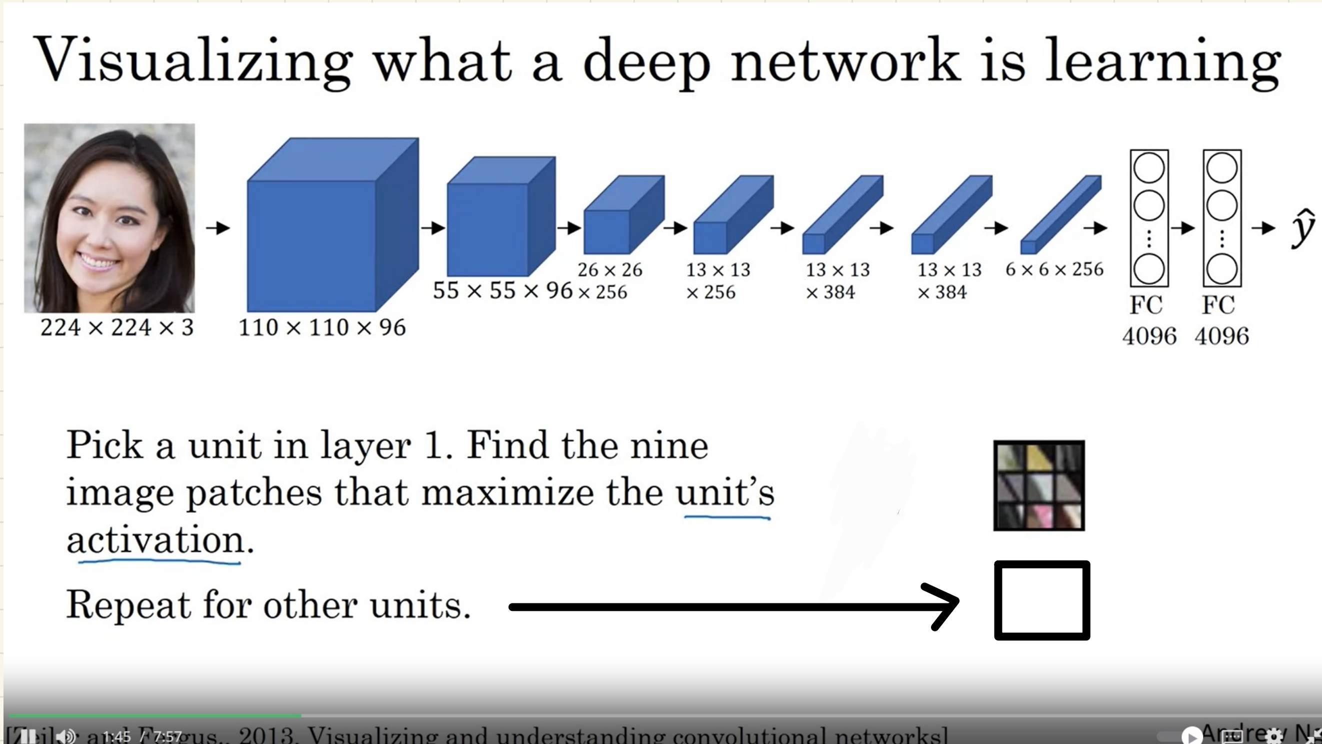
- una vez entrenada la red se puede usar con ejemplos nuevos (one shot learning)

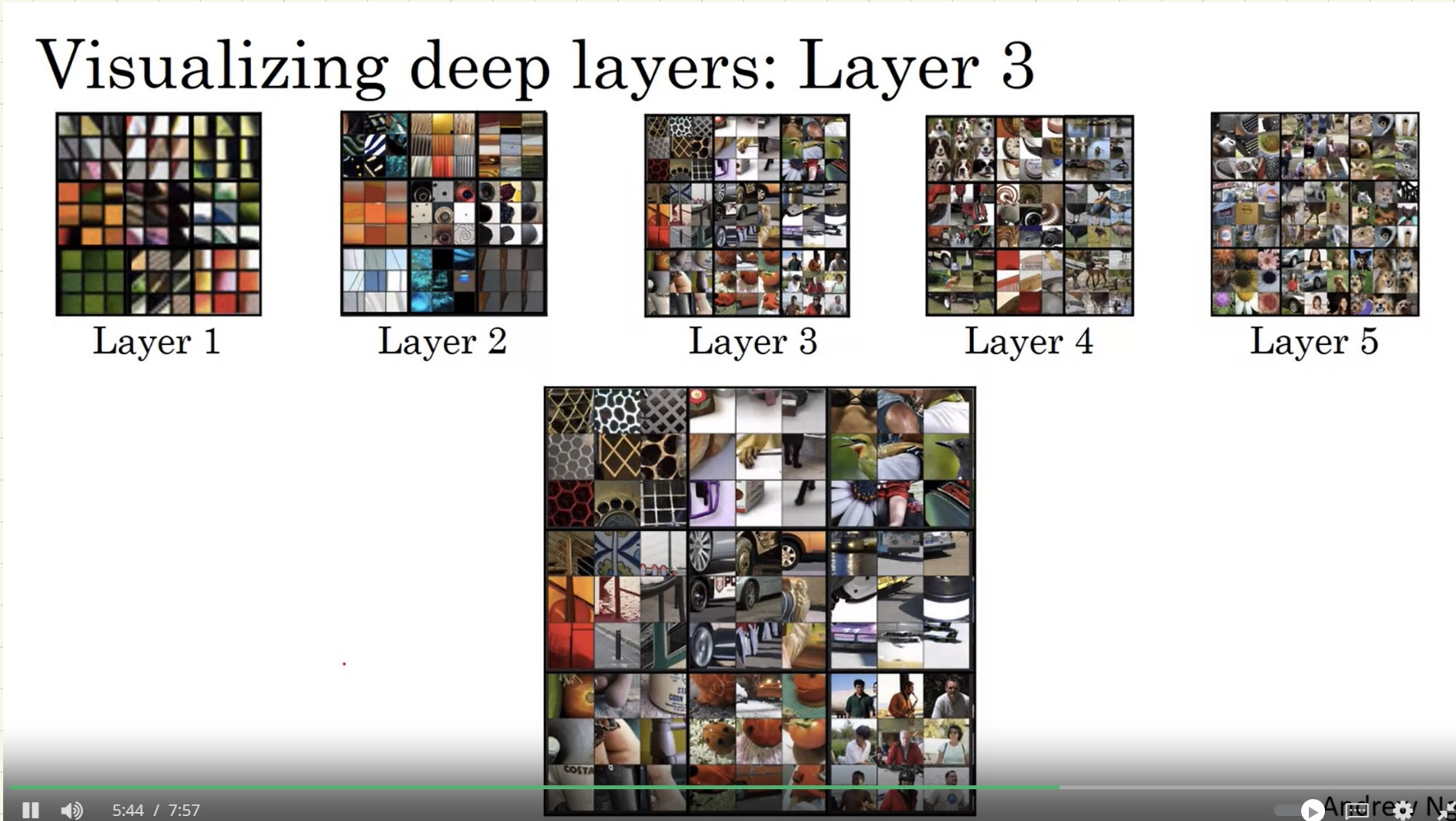
- como un problema de clasificación estándar:



- ¿Qué hace una conv NN?

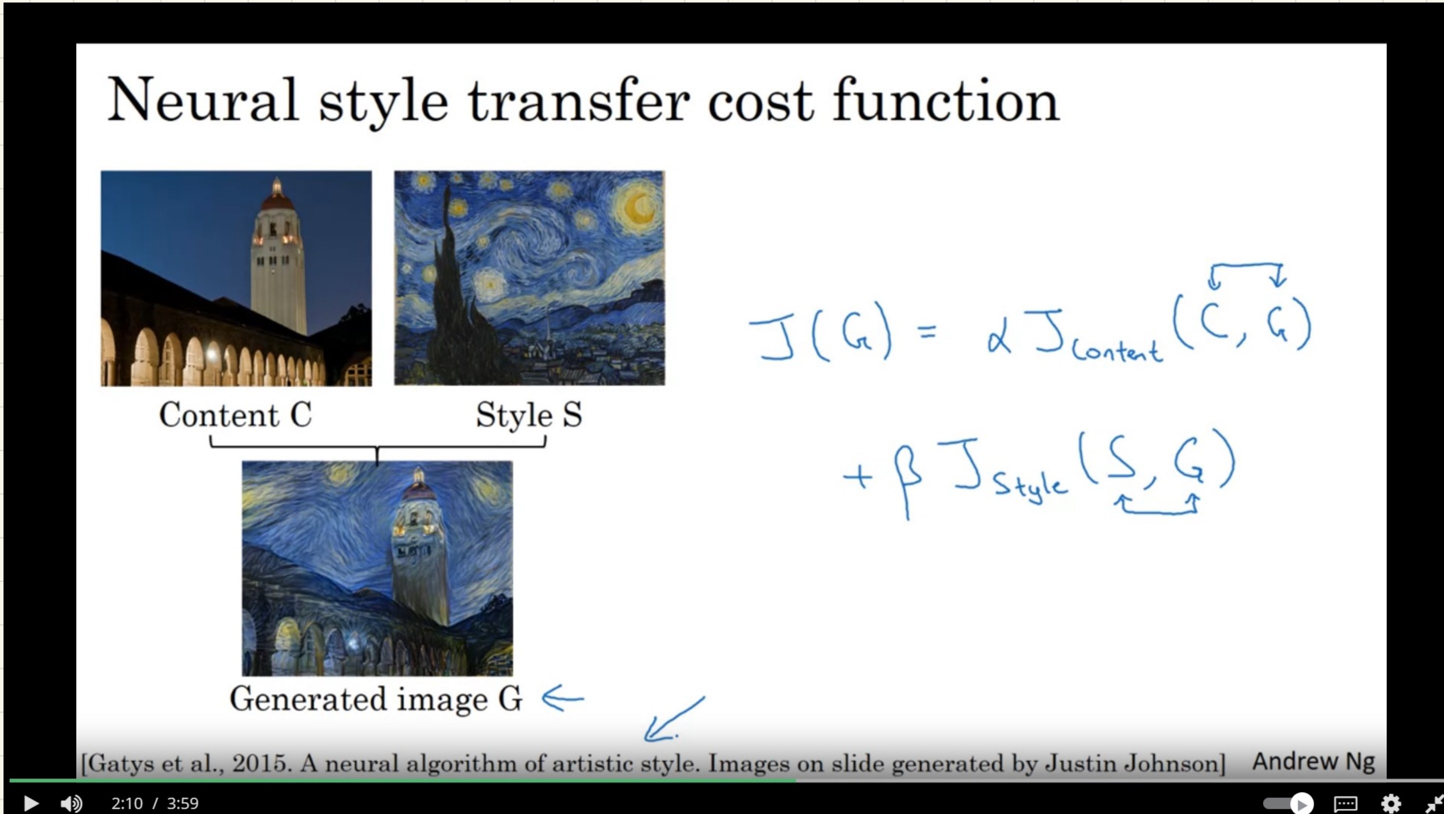
Visualizando qué hace una conv NN: visualizing and understanding conv NN 2013.





Visualizing deep layers: Layer 3
Layer 1
Layer 2
Layer 3
Layer 4
Layer 5
Andrew Ng
5:44 / 7:57

Neural Style Transfer



Find the generated image G

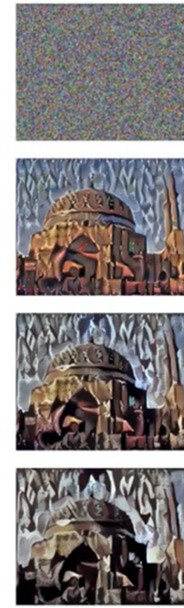
1. Initiate G randomly

   G: $100 \times 100 \times 3$

   RGB

2. Use gradient descent to minimize $J(G)$

$$G := G - \frac{\partial}{\partial G} J(G)$$



[Gatys et al., 2015. A neural algorithm of artistic style]

Andrew Ng

FUNCIONES DE COSTO: contenido

- $J_{content}(C,G)$ se define comparando la representación de C y G en la capa l de una red preentrenada (e.g., VGG)
- Si las funciones de activación son: $a^{[l](C)}$ y $a^{[l](G)}$:

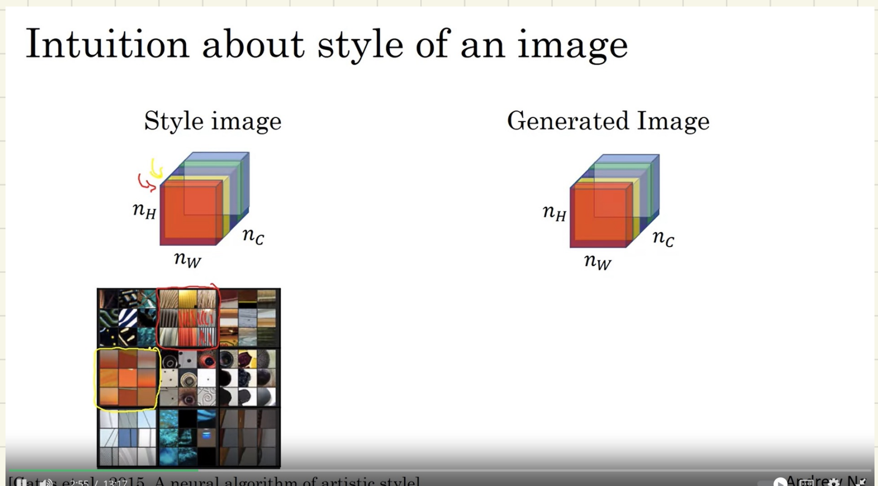
$$J_{content}(C,G) = \frac{1}{2}\| a^{[l](C)} - a^{[l](G)} \|^2$$

FUNCIONES DE COSTO : ESTILO

- primero definimos la correlacion entre canales:

$$G^{[l](G)}_{kk'} = \sum_{i}^{n_H} \sum_{j}^{n_W} a^{[l](G)}_{i,j,k} \, a^{[l](G)}_{i,j,k'}$$

ESTO CAPTURA LA <u>NOCION DE ESTILO</u> caracteristicas que ocurren en diferentes partes de la imagen



- FUNCION DE COSTOS DE ESTILO EN CAPA l:

$$J^{[l]}_{style}(S,G) = \frac{1}{2 n_H^{[l]} n_W^{[l]} n_C^{[l]}} \sum_{k} \sum_{k'} \left(G^{[l](S)}_{kk'} - G^{[l](G)}_{kk'} \right)^2$$

- FUNCION DE COSTOS DE MULTIPLES CAPAS:

$$J_{style}(S,G) = \sum_{l} \lambda^{[l]} J^{[l]}_{style}(S,G)$$

hiperparámetros