

# AI Development Expert

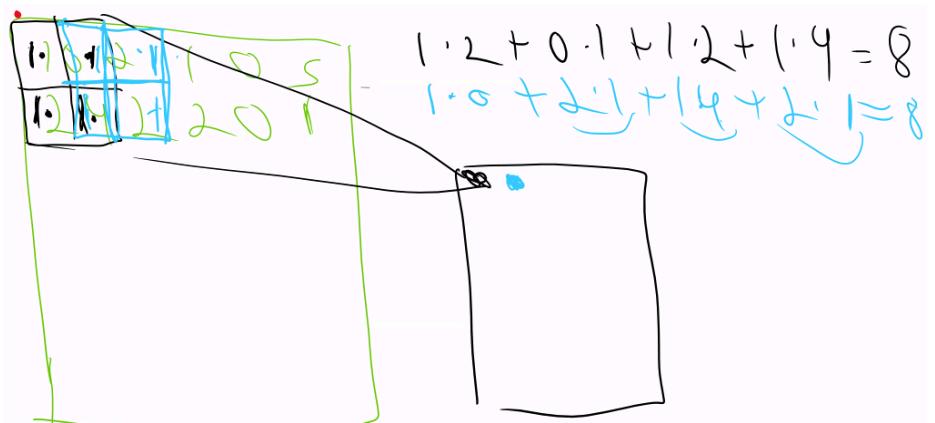
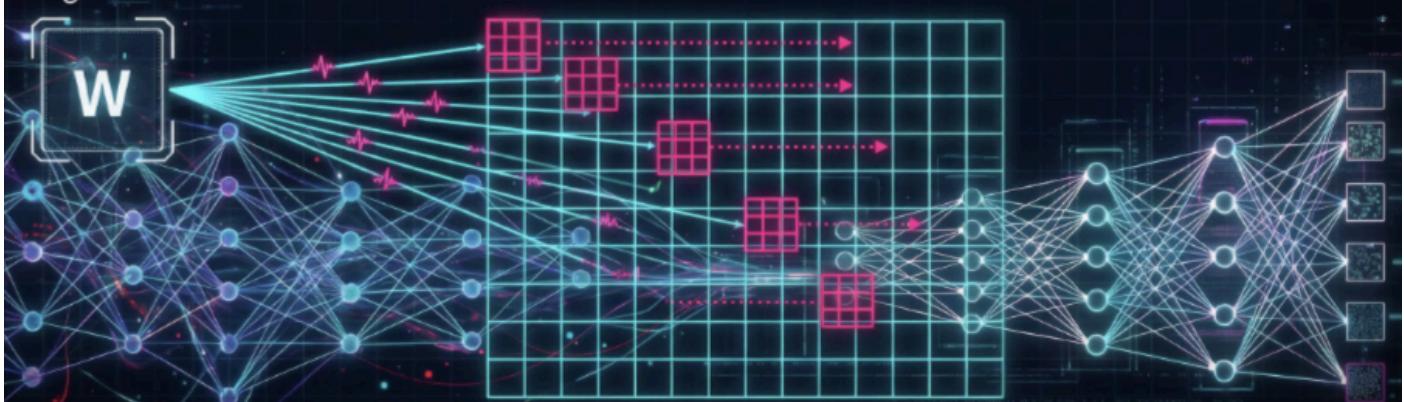
Limudey Hutz | ICCP & ELTA (IAI)

CNN - קונבולוציה

## עקרון שיתוף המשקولات

- אותו פילטר סורק את כל התמונה
- זיהוי בפינה אחת = זיהוי בכל מקום
- חישוכן אדיר בפרמטרים (מאות במקומות מיליוןים)
- אי-שוניות להזזה (Translation Invariance) (Translation Invariance)

Weight Matrix



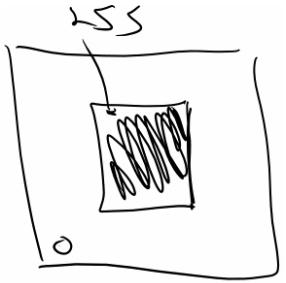
המסיקה של הקרןל - במקומ 1,1,1,1,1 אפשר לשים כל מספר, למשל:

$$\begin{array}{|c|c|} \hline 1 & 1 \\ \hline 1 & 1 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|} \hline w_0 & w_1 \\ \hline w_2 & w_3 \\ \hline \end{array} \Rightarrow \begin{array}{|c|c|} \hline 0 & 2 \\ \hline 5 & -8 \\ \hline -8 & 0 \\ \hline \end{array}$$

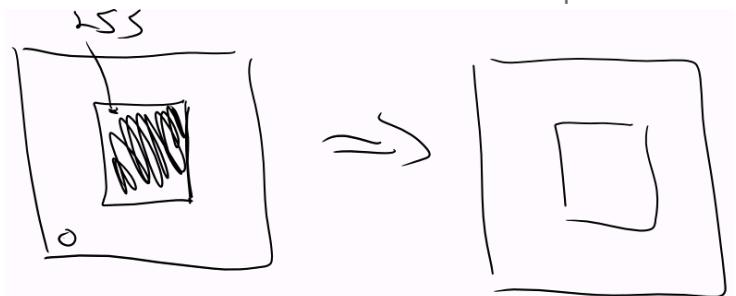
$$w_0 \cdot p_1 + w_1 \cdot p_2 + w_2 \cdot p_3 + w_3 \cdot p_4$$

כלומר יש פה מכפלה סקלרית.  
פעולה הקונבולוציה היא מכפלה סקלרית.

נניח שיש לנו תמונה של קובייה עם צבע אחיד 255 ורקע 0:



המידע יושב בקצוות:



$$\boxed{-1 \ 2 \ -1} = 0 \cdot (-1) + 0 \cdot 2 + 0 \cdot -1 = 0$$

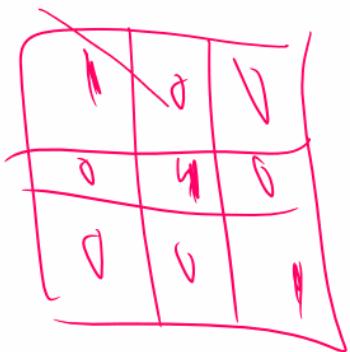
$$\boxed{-1 \ 0 \ 1} = 1 \cdot (-1) + 1 \cdot 0 + 1 \cdot 1 = 1$$

$$\boxed{-1 \ 1 \ -1} = 1 \cdot (-1) + 1 \cdot 1 + 1 \cdot -1 = 0$$

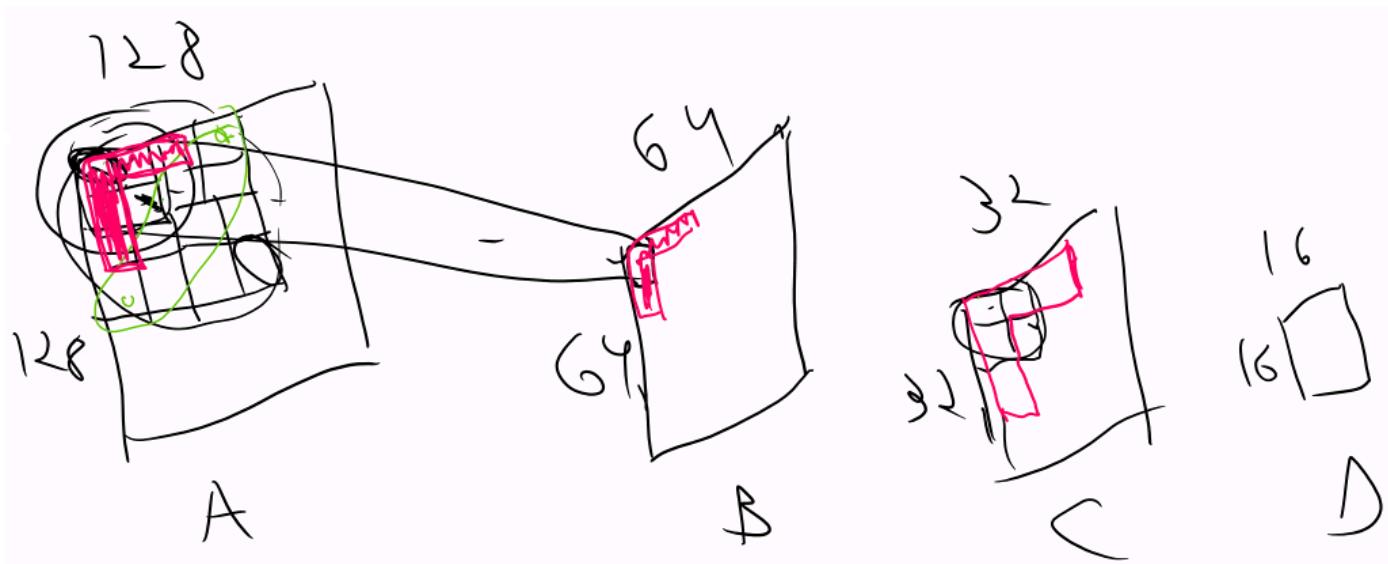
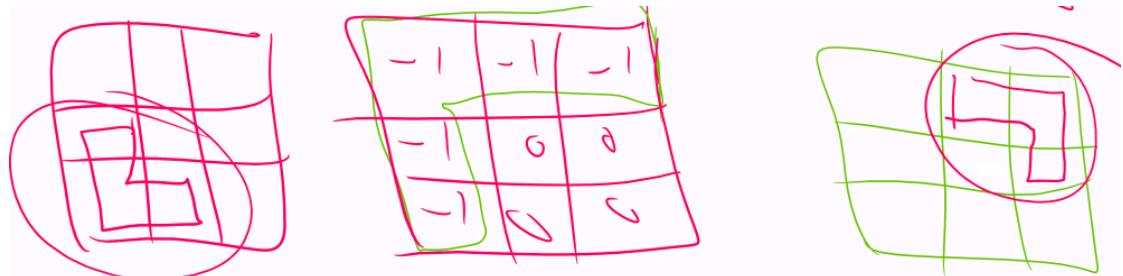
כלומר בכל מקום בתמונה יהיה אפסים, למעט בקצוות.  
זה קרナル ציווני — יודעים רק בשמאלי ימין — לא למעלה ולמטה.  
לכן כדאי לעשות קרナル כזה שיגלה את כל הקצוות:

-1	-1	-1
-1	9	-1
-1	-1	-1

יש גם קernal שינגלה קווים אלכסוניים:

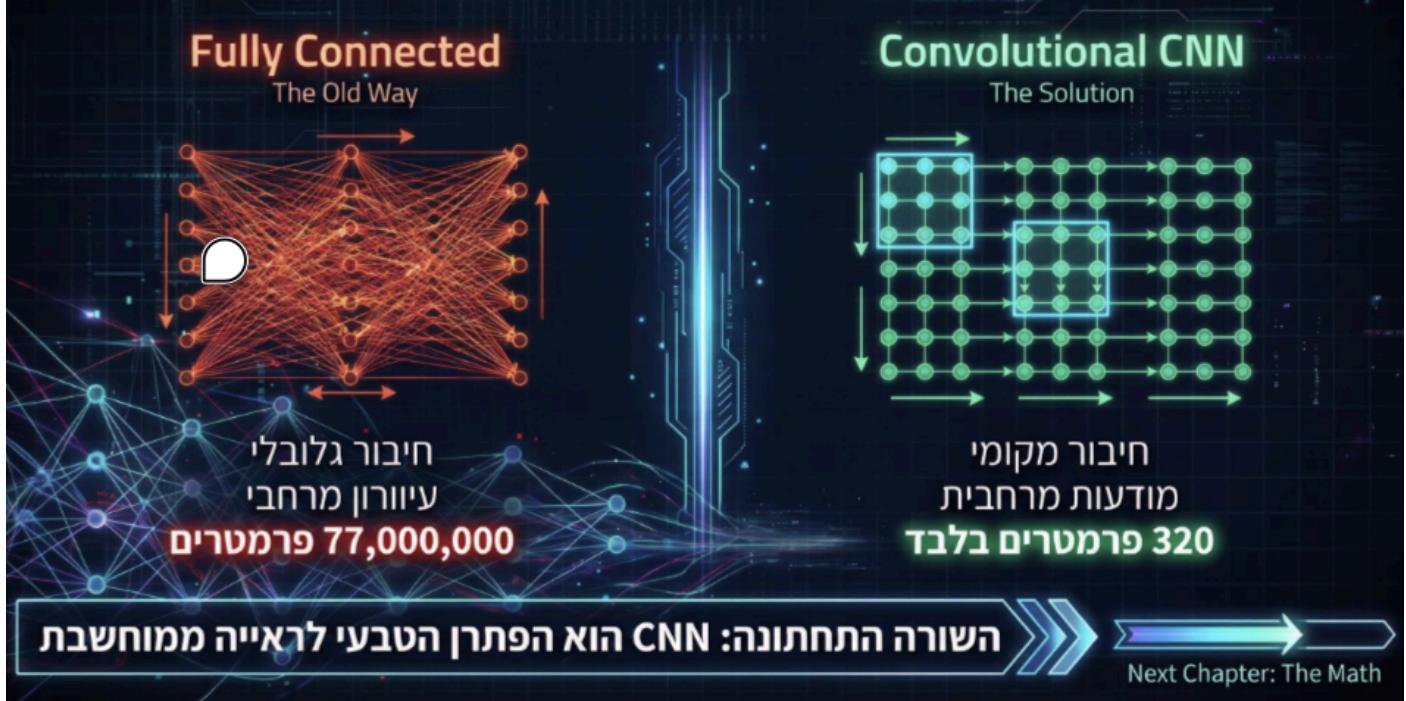


ולכן הקernal הוא כלי לחיפוש צורות בתמונה, במרחב.  
למשל חיפוש צורת "R" או "L":



בכל מקום אני מhapus את אותה הצורה, אבל בהבדל של רגולציה.  
ב- A יש חיפוש של צורות קטנות בצורה הזאת, בשכבה השנייה B אני מhapus את אותה הצורה בגודלים בינוניים, וב- C אני מhapus את הצורה בצורות גדלות - יותר גלובליות. חיבים לעבור את השבלים בכך — אני לא יכול לקפוץ ישר מ A ל- C.  
אני מhapus בהתאם לשאלת המחקר.

## סיכון: FC לעומת CNN



הקרנל מחפש צורה מסוימת. בכל מקום שאין את הצורה הפיסקלים יקבלו מספרים קטנים, במקומות שיש הפיקסל קיבל ערך נזול.

# דוגמה מספרית: קונבולוציה 1D

חיפוש דגם באוט

Signal ( $f$ )



LOCK ON

$$1 \cdot 1 + 2 \cdot 2 + 3 \cdot 3 + 1 \cdot 1 = 15$$

Filter ( $g$ )



Output Map

15

15

## הגדרה הפורמלית - קונבולוציה 1D

$$(f * g)(t) = \sum f(\tau) \cdot g(t - \tau)$$

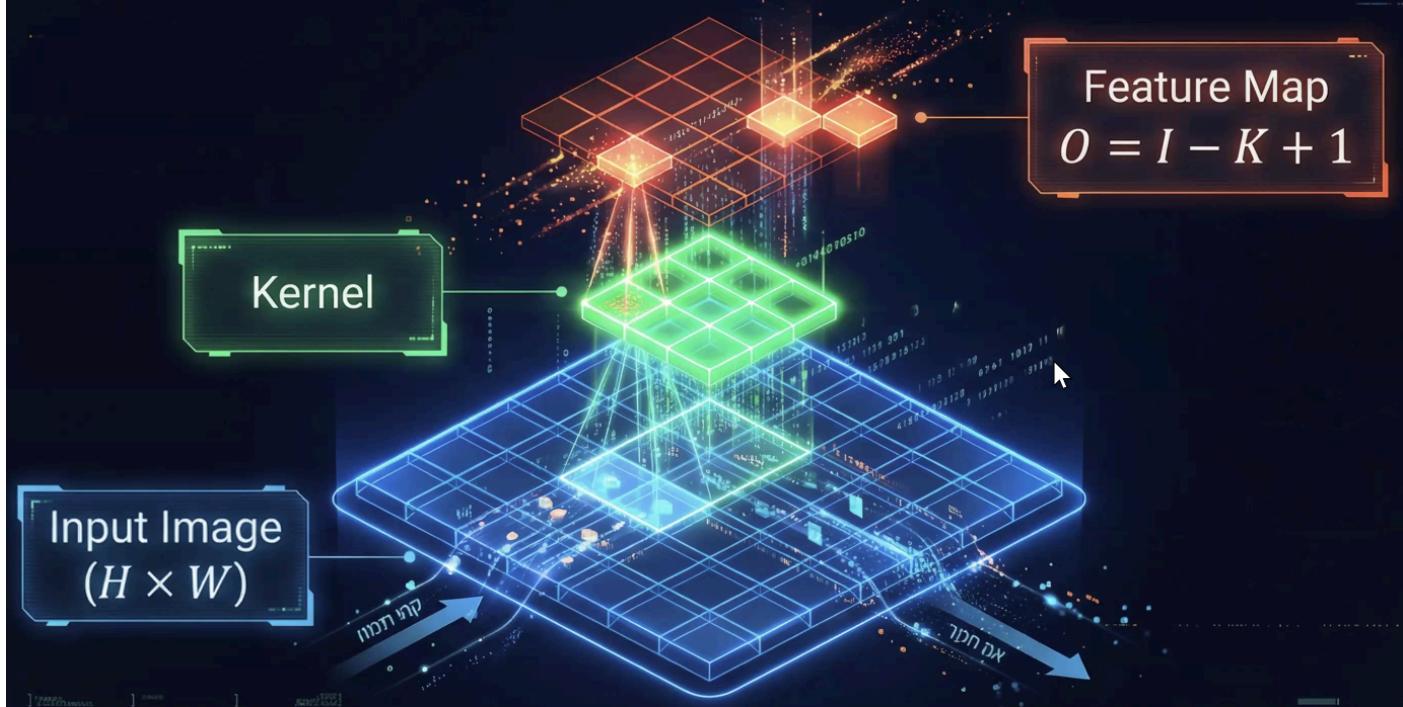


היפוך המסנן

הבדל מקורסיה: המסנן מתרافق לפני הזרזה

# קונולוציה דו-ממדית

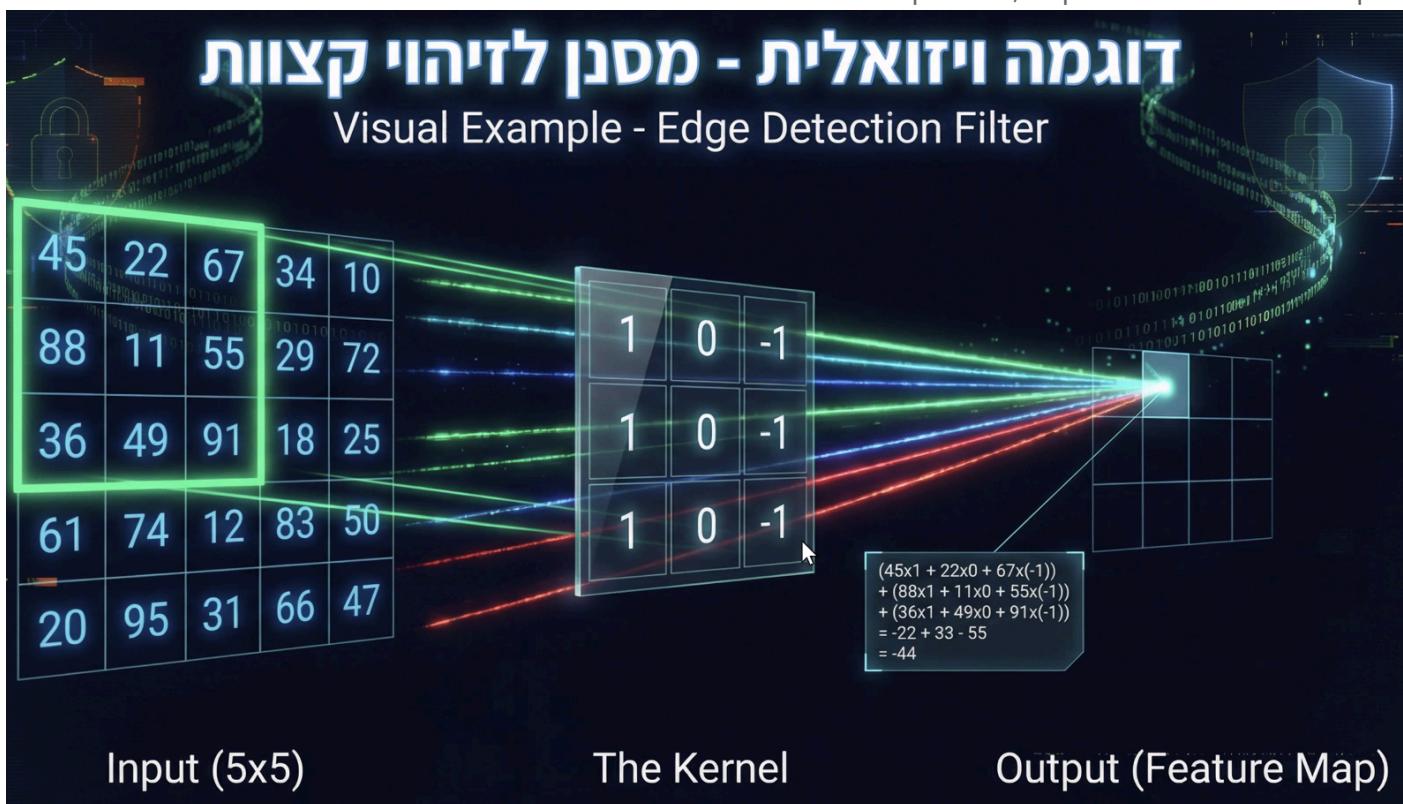
## מאtotות לתמונות



W - זה ה Kernel

H - זה התמונה

במקום שיש את הצורה של הקernel, יש חיזוק בתמונה.



קרナル בהנדסה יהיה מספר או-זוני של מספרים, כי מוחפשים את נקודת המרכז.

נגיד לוקח 9 שכנים - ובסוף נקודות המרכז תתקבל -44.

פעולת קונולוציה משנה את התמונה! תהיה עכשו תמונה של פיקסלים חדשים שמחזקים מידע שעונה על שאלת המחקר וממניכים מה שלא עונה על שאלת המחקר.

# קונבולוציה מול קורלציה

הבדל: היפוך: The Difference:

## Correlation



Kernel slides \*without\* orientation change.

## Convolution



Kernel \*rotates 180°\* before sliding.



Practical Reality  
In Deep Learning: Weights are Learned → Difference is Irrelevant



קורלציה — לא מבצעת את היפוך.

קונבולוציה — כן מבצעת את היפוך.

מי שמחלית על המשקלים — זה הרשות.

מה שהולכים לאמן זה בעצם המשקלים של הקרNEL.

בקונבולוציות Neural Network מאמנים את הרשות למשקלים של הקרNEL.

הרשות מוצאת את הקרNELים שייענו על שאלת המחקר.

ה- **Fully Connected** — עובדת על הסתברויות.

ה- **CNN** — תחליש מה שלא מתאים לKERNEL, ותזקק מה שמתאים לKERNEL.

## תכונות אלגבריות

הכוח המתמטי של הקונבולוציה

### חילופיות

$$f * g = g * f$$

### קיבוץיות

$$f * (g * h) = (f * g) * h$$

איחוד מסכניםים

### lienarיות

$$\text{Superposition}$$

### הזהה

$$\text{Shift Input} \rightarrow \text{Shift Output}$$

הזהה — אם אוזע את כל התמונה ימינה, אז גם תוצאה הKERNEL תזוז ימינה, בשונה מ- fully connected שזהה עלולה לשבעש לנו MRI את ההחלטה האם זה חתול או לא.

# קונבולוציה כמכפלת מטריצות

מטריצת טפליז והאצת חומרה

The Math - Toeplitz Matrix

The Hardware - GPU

1	2	3	4	5	6	1	1	1
3	10	2	3	5	3	3	2	1
4	3	10	2	3	6	3	3	2
6	1	2	10	2	3	6	5	5
5	6	1	2	10	2	3	6	3
6	5	6	1	2	10	2	3	3
3	4	5	6	1	3	10	3	3
1	2	3	4	6	1	3	10	2
1	2	3	4	5	6	1	2	10

Toeplitz Matrix

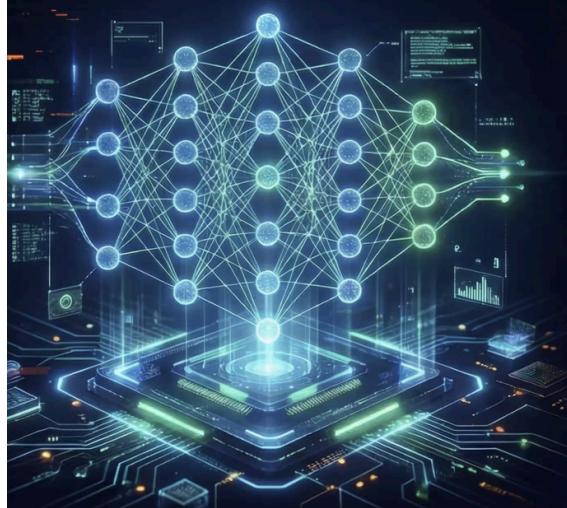


GPU Acceleration (cuDNN)

קונבולוציה מהוות מכפלת מטריצות.  
GPU משמש נולץ לביצוע קונבולוציות.

## סיכום ומסקנות

מה אינטואיציה למימוש



✓ קונבולוציה = חיפוש צורות (הבלש)

✓ הגדרה מתמטית: D1 ו-D2

✓ קונבולוציה או קורלציה: היפוך

✓ תכונות: חילופיות, לינאריות

✓ ייצוג מטריצי: יעילות ב-GPU

הצעד הבא: כיצד מחשב 'רואה' תמונה? (פרק 3)

# ייצוג תמונות דיגיטליות ועיבוד מקדים

Digital Image Representation and Preprocessing

## מפוטונים לפיקסלים

כיצד מחשב רואה תמונה?



ראייה אנושית

0	0	0	0	0	0	0	0	0	0	0
0	0	11	12	91	97	164	80	35	85	
0	18	76	41	86	192	166	184	154	95	
0	45	45	62	85	102	138	160	152	125	
255	72	95	59	31	99	157	152	170	235	
255	170	102	102	81	85	123	189	198	235	
0	32	48	69	76	182	158	248	215	255	
0	15	33	55	88	132	162	193	231	255	
0	0	15	33	75	123	135	219	232	255	
0	0	0	0	00	136	157	200	255	255	

ראייה ממוחשבת

תמונה דיגיטלית = מטריצה של מספרים

# תמונה רמת אפור – מערכים דו-ממדיים

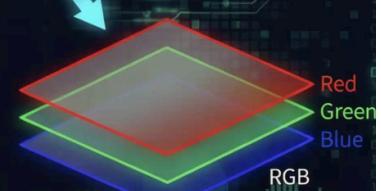
0	0	16	90	255
0	45	128	200	255
0	77	128	210	255
0	128	142	215	255
0	40	87	42	0

Zoomed-in Image Patch (5x5 Grid)

$$I \in \mathbb{R}^{H \times W}$$



1 Channel



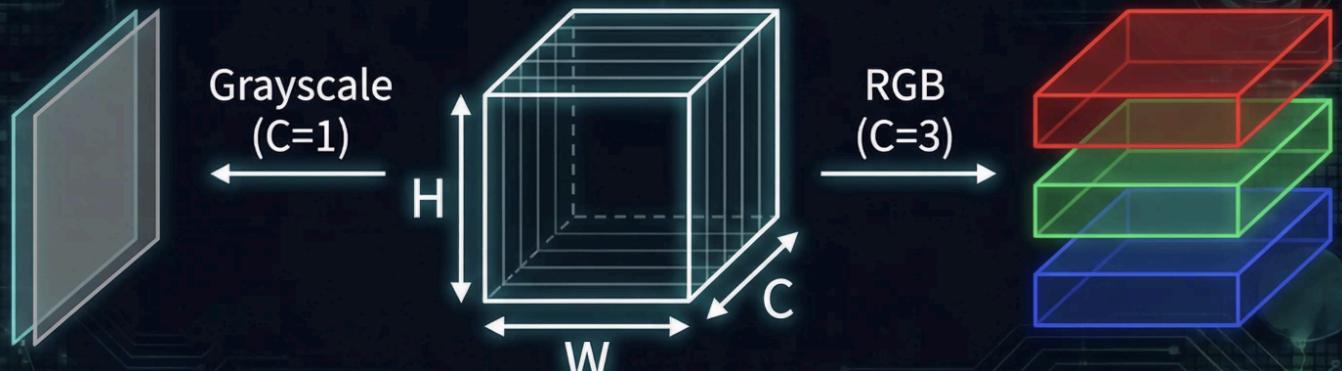
3 Channels

כל פיקסל = ערך בודד של עצמת אור  
טווח: 0 (שחור)  $\leftrightarrow$  255 (לבן)

יתרונות: פשוטות ויעילות חישובית

## (H, W, C) -قطנותור -

$$I \in \mathbb{R}^{H \times W \times C}$$



Batch (N, H, W, C)

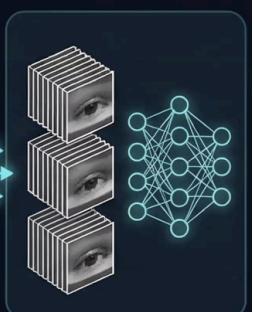
- אוסף תמונות = אצווה (Batch)

# 齊納ור העיבוד המקדים

**קלט (Input)**  
תמונה גולמיות



**פלט (Output)**  
טנזוריים מעוצבים



ככל שהדadata נקי יותר, התוצאות טובות יותר

## אסטרטגיית נורמל

התאמת הנתונים לדרישות הרשת

**Min-Max [0,1]**



➡ Formula:  $\hat{I} = I / 255$

Usage: Default (ברירת מחדל)

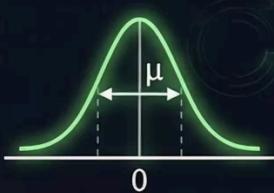
**Min-Max [-1,1]**



➡ Formula:  $\hat{I} = \frac{I}{127.5} - 1$

Usage: GANs, Tanh

**Z-Score**



Formula:  $\hat{I} = \frac{I - \mu}{\sigma}$

Usage: Standardization

**Per-Channel**



Usage: ImageNet, Color

# הרחבת נתונים (Data Augmentation)

шиיפור הcalculה ע"י ויריאציות

סיבוב (Rotation)



זום (Zoom)



הזזה (Shift)



שיקוף (Flip)



תמונה מקורית

**זהירות:** שמירה על משמעות סמנטיבית!

## סיכום: מהפיקסל לרשט

סיכום: מהפיקסל לרשט



**Digitization:**

דגימה + קוונטיזציה



**Representation:**

טנזור ( $H, W, C$ )



**Preprocessing:**

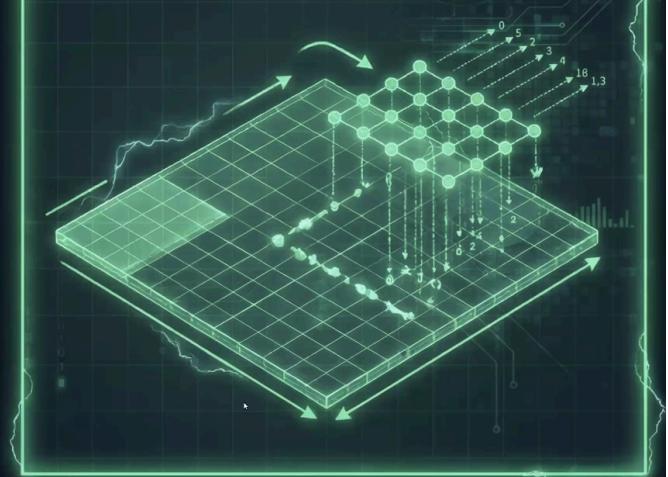
Resize → Normalize → Reshape



**Augmentation:**

הרחבת נתונים

**בפרק הבא: קונבולוציה**



# גרעינים, מסננים וריפוד

## Kernels, Filters, Padding and Stride

### מבוא היסטורי - המיצאת גרעין Sobel



- 1968: Gary Feldman ו Irwin הציגו מטריצה זעירת של  $3 \times 3$
- היכולת 'לראות' קצוות באמצעות 9 מספרים בלבד
- 50 שנה מאוחר יותר: הבסיס לרשתות קונבולוציה (CNN)
- המהפהכה: הרשת לומדת את המספרים עצמה

### מהו גרעין? - הגדרה פורמלית



- **אנלוגיה:** חוקר עם זכוכית מגדלת הסורק את התמונה



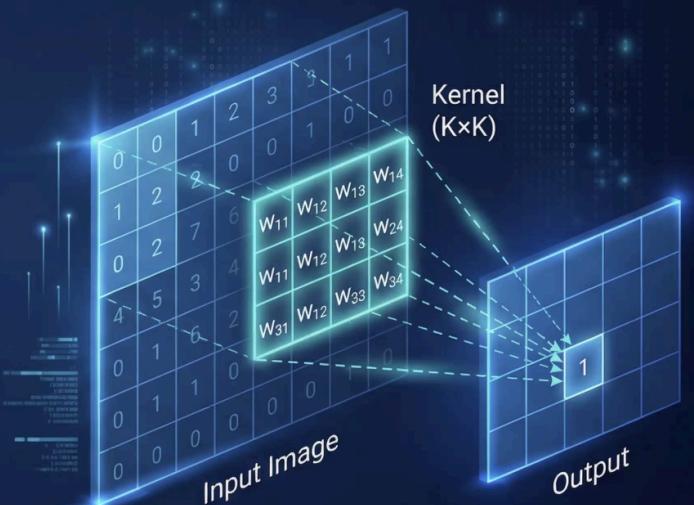
- **הגדרה:** מטריצה קטנה ( $K \times K$ ) המכילה משקלות



- **פעולה:** מכפלה נקודתית (Convolution) וסכום (Summation)



- **תובנה:** הגרעין הוא המשקלות של הנירון



$$O(i, j) = \sum_{i=1}^n \sum_{j=n}^n I(i + m, j + n) \cdot K(m, n)$$

# גרעינים קלאסיים - Prewitt ו-Sobel

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

(אופקי) Sobel-x

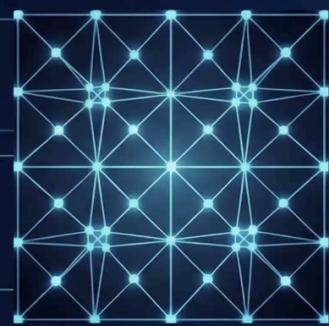
$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

(אנכי) Sobel-y

זיהוי קצוות: הצעד הראשון בהבנת תמונה  
סימטריה: הפרש בין צדדים מנוגדים  
דומה ל-Sobel אך עם ערכי  $\pm 1$ : Prewitt

## גרעינים נוספים - שטוש וחידוד

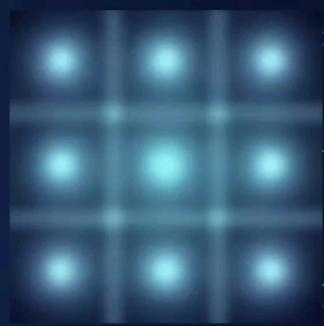
### חידוד (Sharpen)



$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

מחזק הבדלים - מדגיש פרטים עדים

### שטוש (Blur)



$$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

ממוצע של פיקסלים שכנים - מחליק את התמונה

תובנה: גרעינים קלאסיים אלו עוצבו ידנית על ידי מהנדסים

# גרעינים נלמדים - המהפהכה של CNN

From Hand-Designed to Learnable Parameters



מתחלים במספרים אקריאים בקורסן, ובסוף אנחנו רוצים לענות על שאלת המחוקך.  
למשל האם בשיטת הזום - כל פיקסל הוא דמות אדם או רקע.

## שיטת משקלות - הפטנט המרכזי

- **הרעין:** כל הנירונים המפעילים את הגרעין משתמשים באותו ערכיהם

- **יעילות דרמטית:**
  - גרעין  $3 \times 3 \leftarrow 10$  פרמטרים בלבד (כולל Bias)
  - גרעין  $5 \times 5 \leftarrow 26$  פרמטרים בלבד

- **השוואה:** ברשת FC כל נירון לומד משקלות נפרדים

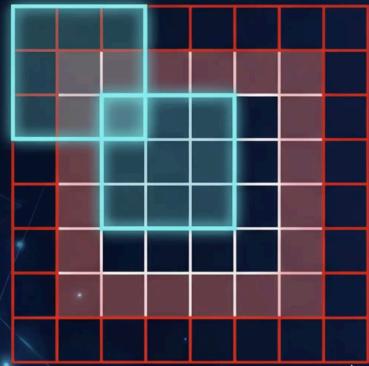
- **齊וטו:** 'המשקל  $k_1$  זהה לו אותו משקל  $k_1$  שם'



# בעיית השולטים וריפורד

## הבעיה: גרעין $3 \times 3$ על תמונה $6 \times 6$

- הגרעין לא יכול להתרחץ על פיקסלים בשוליים
- תוצאה: תמונה הפלט קטנה מהתמונה המקורי



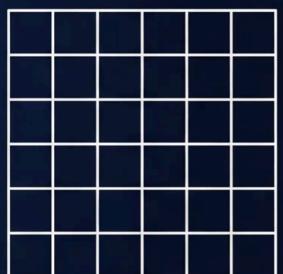
תמונה פלט ( $4 \times 4$ )

## הפתרון: ריפורד (Padding)

- הוספת פיקסלים מלאכותיים (אפסים) מסביב לתמונה
- אפשר לגראן להתרחץ על כל פיקסל הקלט המקורי

0	0	0	0	0	0	0	0
0							0
0							0
0							0
0							0
0							0
0							0
0	0	0	0	0	0	0	0

תמונה קלט עם ריפורד ( $8 \times 8$ )



תמונה פלט ( $6 \times 6$ )

ציטוט: "יש בעיה במסגרת החיצונית"

בעיית השולדים (Padding) — התמונה היא הורוד מצד שמאל. הקermal יתחל מפיקסל מעלה ומצד שמאל לתמונה בשל בעיית השולדים — זה מאפשר לkernel להתרחץ על כל פיקסל הקלט המקורי.

לכן נוצרה להחליט האם לעשות Padding או לא. במקרה הקונבולוציה נשמר על החרולציה המקורי.

## שתי אסטרטגיות ריפורד:

High-Tech Comparison Slide

### Valid Padding (ללא ריפורד)

1	2	3				
4	5	6				
1	1	1				

Input ( $5 \times 5$ )



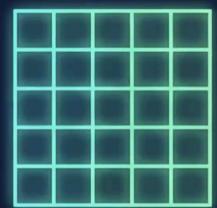
Output ( $3 \times 3$ )

הגרעין גולש רק בתחום הגבולות  
ה 结果: הפלט מתכווץ  
נוסחה:  $O = I - K + 1$

### Same Padding (ריפורד באפסים)

0	0	0	0	0	0	0
0						0
0						0
0						0
0						0
0						0
0	0	0	0	0	0	0

Input ( $5 \times 5$ )

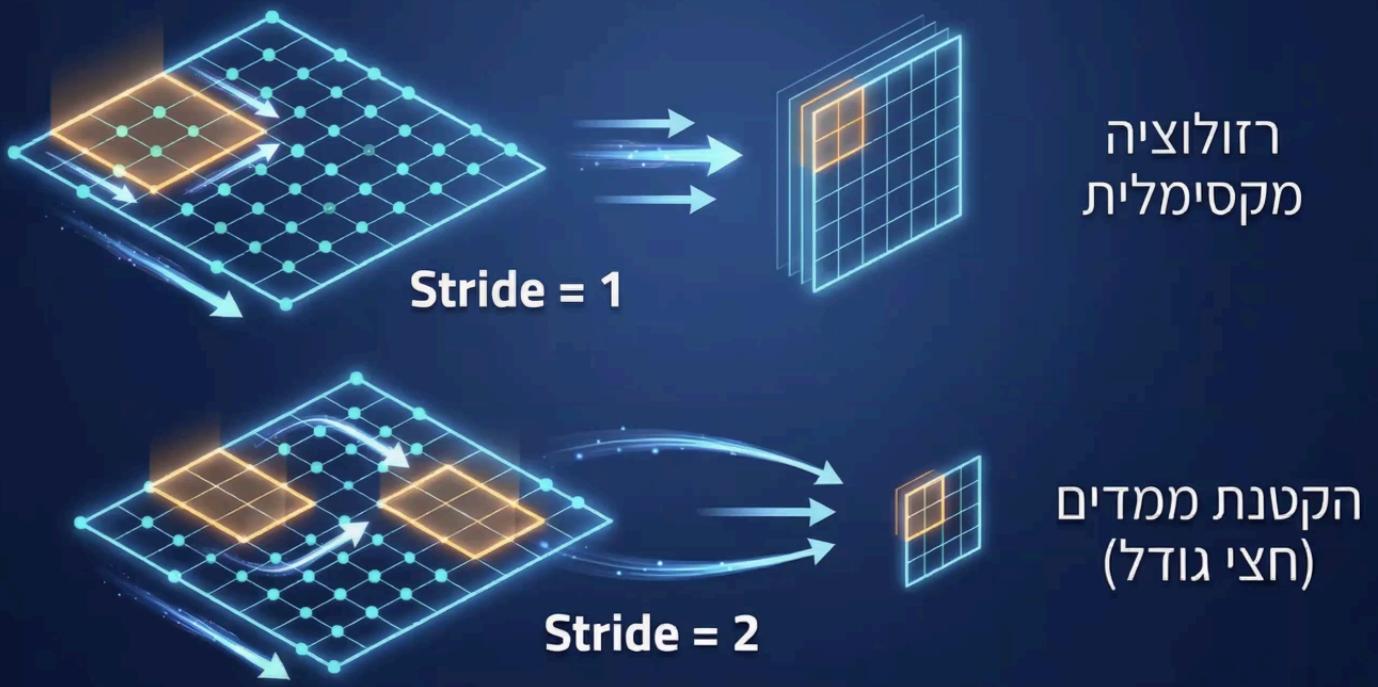


Output ( $5 \times 5$ )

הוספת מסגרת אפסים  
ה結果: גודל הפלט = גודל הקלט  
נוסחה:  $P = \lceil K/2 \rceil$

"שכבת קונבולוציה קלאסית שומרת על גודל תמונה המקורי בזכות הריפורד"

# צעד הגרעין - Stride



- **הגדלה:** המרחק שהגרעין מתקדם בכל צעד
- **הפשרה:** מהירות חישוב (Speed) לעומת פירוט (Detail)
- **בפרקטייה:** לרוב משתמשים ב-Stride=1 לשימור מידע

בעיקרון קונבולוציה לא תנסה (אם פдин) את הריזולוציה.

למה ארצתה להקטין ריזולוציה של תמונה — כדי לזרות צורות בגודלים גדולים יותר.

מצד אחד מתחשים צורה, ומצד שני אני מקטין את התמונה כדי שכל פעם אסתכל בסקללה אחרת.

1- גילוי הצורה == פועלות קונבולוציה

2- הקטנת ריזולוציה == נדרש **resizer**. לוקחים קרnel של פיקסלים וופר אותם לממוצע שלהם. אם לקחת 4 פיקסלים אז

בבת אחת ייצורתי מזה פיקסל אחד.

בכמה להזיז את הקרnel - זה מה ש stride עושה.

תמונה עם  $32 \times 32$ , גודל הkernel  $2 \times 2$ , פסיעה = 2. לכן היחסית תהיה חצי:

### 1. דיהו הנתונים

- רוחשיות כניסה ( $I$ ):  $32 \times 32$
- גודל הקרנל ( $k$ ):  $2 \times 2$
- פסיעה ( $s$  או  $stride$ ): 2
- ריפוד ( $padding$  או  $p$ ): 0 (בנהча שלא צין אחרת)

### 2. הנוסחה לחישוב ממד הפלט

כדי למצוא את גודל הפלט ( $O$ ). משתמש בנוסחה הסטנדרטית המשמשת בשכבות קונבולוציה או Pooling:

$$O = \lfloor \frac{I - k + 2p}{s} \rfloor + 1$$

### 3. הצבת הנתונים וחשבון

ציבב את הערכים שבירידים:

$$O = \lfloor \frac{32 - 2 + 2(0)}{2} \rfloor + 1$$

$$O = \lfloor \frac{30}{2} \rfloor + 1$$

$$O = 15 + 1 = 16$$

## נוסחת גודל הפלט: השוויון המרכזי

### Output Size Formula: The Central Equality

**מרקם משתנים:**  
I = גודל הקלט  
K = גודל הגרעין  
P = ריפוד  
S = צעד  
O = גודל הפלט

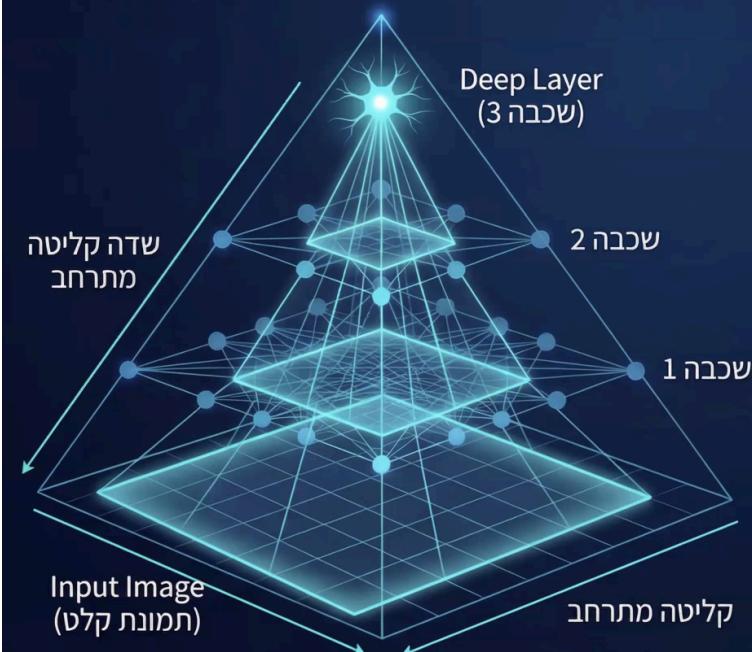
### דוגמאות:

1. תקף:  $28 \times 28, K = 3, P = 0, S = 1 \rightarrow O = \left\lfloor \frac{28 - 3 + 2(0)}{1} \right\rfloor + 1 = [25] + 1 = 26$

2. זהה:  $28 \times 28, K = 3, P = 1, S = 1 \rightarrow O = \left\lfloor \frac{28 - 3 + 2(1)}{1} \right\rfloor + 1 = \left\lfloor \frac{27}{1} \right\rfloor + 1 = 28$

3. צעד:  $28 \times 28, K = 3, P = 1, S = 2 \rightarrow O = \left\lfloor \frac{28 - 3 + 2(1)}{2} \right\rfloor + 1 = \left\lfloor \frac{27}{2} \right\rfloor + 1 = [13.5] + 1 = 13 + 1 = 14$

# שדה קליטה: כמה מהתמונה רואה כל ניורון?



החותמים:

- האзор בתמונה המקור שמשפיע על ניירון מסוים

משונה:

$$R_L = 1 + L \cdot (K - 1)$$

משכנה:

$$3 \text{ שכבות של } 3 \times 3 \rightarrow \text{שדה קליטה } 7 \times 7$$

שכבות עמוקות רואות אזור רחב יותר  
→ זיהוי דפוסים מורכבים

## גדלי גרעין נפוצים

גודל	שימושים (Use Cases)	פרמטרים (Params)	שדה קליטה (Receptive Field)
1x1	שילוב ערכאים	2	1x1
3x3	הנפוץ ביותר	10	7x7
5x5	דפוסים גדולים	26	13x13
7x7	שכבה ראשונה	50	19x19



Two 3x3 layers = Same Receptive Field, Fewer Parameters (20 vs 26)

## סיכום: אבני הבניין של שכבת קונולוציה

**גרעינים נלמדים:**  
הראש לומדת את המספרים בלבד (במקום חכון ידני)



**גרעין/MSN:**  
מטריצה קטנה שגולשת על התמונה  
ומבצעת מכפלה סקלרית.



**ריפוד (Padding):**  
שימוש ב-Padding Same שומרה על גודל התמונה.



**שיטוף משקלות:**  
אותו גרען רץ על כל התמונה.  
← חיסכון דרמטי בפרמטרים.



**שדה קליטה:**  
שכבות עמוקות רואות חלק  
גדול יותר מהתמונה.



**Stride:**  
שליטה ברוחולוציה (צעד גדול =  
פלט קטן).



הצעד הבא: מפות פיצרים (Feature Maps)

## פרק 5: כוחה של המקבילות

### מטרה: ד"ר יורם סגל



הבעיה

פנס בודד = מסנן בודד



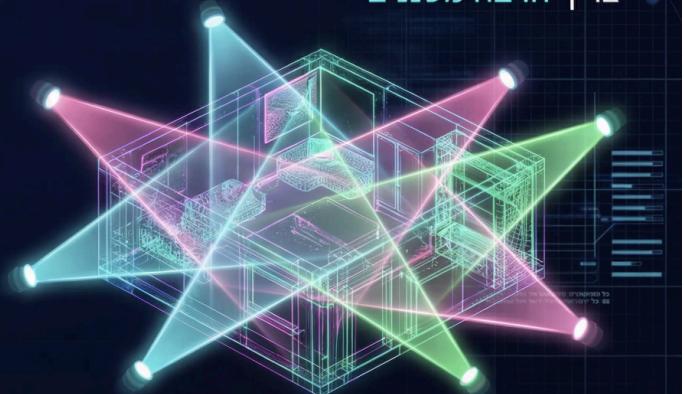
אנלוגיית הפעס: למה צריך הרבה?

- פנס בודד מאיר רק פינה אחת בלבד

- מסנן בודד 'ראה' רק תבנית אחת

- כדי לראות את השלם (קצוות, פינות),

צריך הרבה מסננים



הפתרון

הרבה פנסים = תמונה מלאה

קרנבל אחד מאיר איזור אחד בחדר. קרנבל אחד רואה תבנית אחת. כדי שנראה את כל החדר, חיברים להפעיל הרבה קרנבלים.

# מסננים מרובים = מומחים שונים



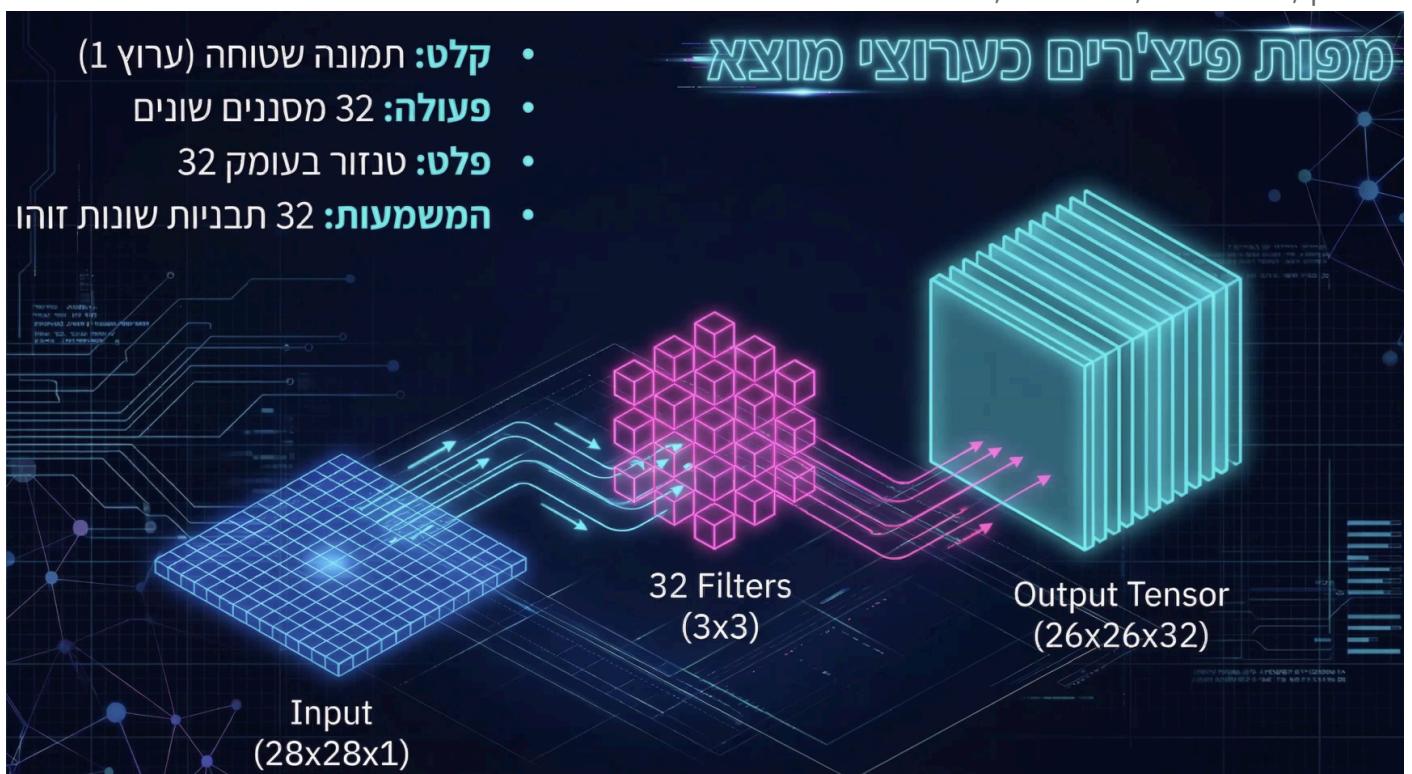
- צוות של 32 מומחים סורקים את התמונה
- מומחה 1: מחפש קצוצות אופקיים
- מומחה 2: מחפש קצוצות אנכיים
- התוצאה: 32 מפות פיצ'רים שונות

כל קרNEL מומחה לבעה אחרת.

אחד לאפ', אחד לעיניים, אחד לנבות, ...

## מפות פיצ'רים כערוצי מואץ

- **קלט:** תמונה שטוחה (ערוץ 1)
- **פעולה:** 32 מסננים שונים
- **פלט:** טנזור בעומק 32
- **המשמעות:** 32 תבניות שונות זוחה



כזכור, מכניםים תמונה אחת אבל מוצאים המונח-המן תמונות: Tensor.

כל תמונה שיוצאה זה פועלה של קרNEL אחד. כל קרNEL מייצר תמונה.

ונוצר טנזור (טאך של תמונות) שכל תמונה מתארת מההו אחרית על הנושא.

בדוגמא, לא השתמשו ב Padding ולכן הרוחולציה ירדה.

## חישוב ממדי הפלט

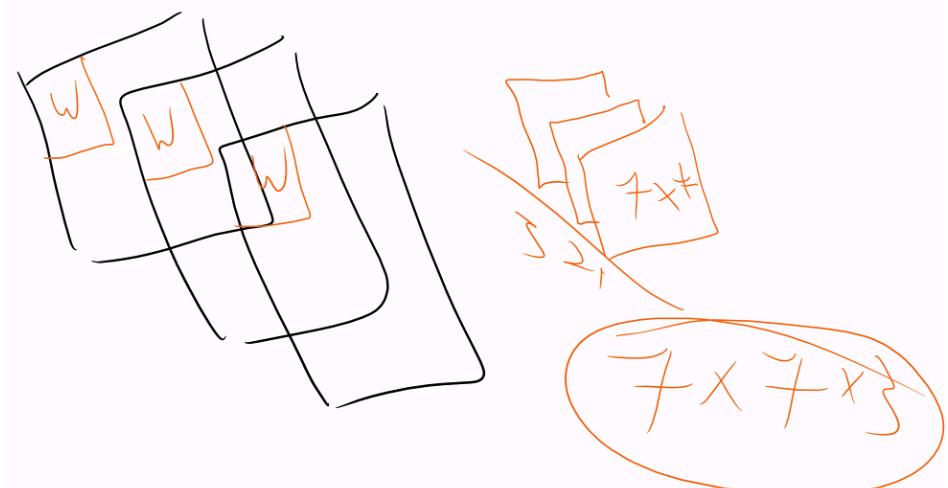
$$H_{out} = \frac{(H_{in} - K + 2P)}{S} + 1$$

$H_{in}$  = 28 גובה קלט  
 $K$  = 3 גודל גרעין  
 $P$  = 0 ריפוד  
 $S$  = 1 פסיעה

$$\begin{aligned}
 &(28 - 3) / 1 + 1 \\
 &25 + 1 = 26 \\
 &\boxed{26 \times 26 \times 32}
 \end{aligned}$$

כמויות השכבות (channels) בטנוור נקבעת על פי כמויות הkernelets.  
גודל הקרNEL נקבע על פי גודל התמונהה בהתחלה.

אם אני עובד על תמונה צבע (בנוייה מ- 3 תМОנות), ואני מפעיל עליה קרNEL. הקרNEL שניי מפעיל על התמונה בכניסה ייצור טנוור של 3 תMONות שעברו קונבולוציה של הקרNELים האלה.



### TASK: [Home Assignment L38]

תרגיל:

לסכם את השיעור.

להשתמש ב- notebook, להשתמש ב- nanobanana לשרטוטים מעניינים.

להסביר בצורה מקיפה וברורה מה זה CNN, מה התהליכיים.

הפעם, הרעיון הוא לא לחת משהו אוטומטי מ LLM, אלא ממש להראות הבנה. יכולם להסביר במילים שילך בן 15 יבין זאת בזדאות. לכתוב זאת במילים ולסכם במילים של!

לחת דוגמאות של תMONות בגודלים שונים עם שכבה אחת או יותר, עם פדיםנסים שונים, עם strides שונים g