Deep Learning

Computer vision

Computer Vision

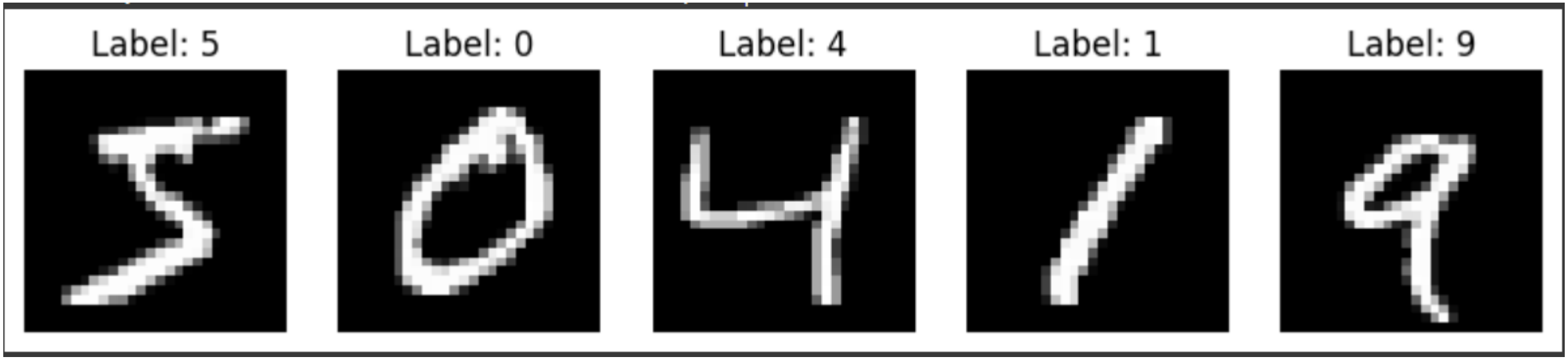
- MINST Data Set
- Flatten Layer
- Convolutional Layer
- Max Pooling
- Data Augmantion

Computer Vision

Computer Vision הוא תחום בבינה מלאכותית המאפשר למחשבים להבין ולנתח תמונות וסרטונים. באמצעות רשתות נוירונים, ניתן לזהות עצמים, לסווג תמונות ולבצע משימות כמו זיהוי פנים, זיהוי תנועה, וראייה רובוטית.

מסדי הנתונים MNIST/Fashion-MNIST

MNIST הוא דאטסט פופולרי של ספרות 9-0 בכתב יד, המשמש לאימון מודלים בסיסיים של למידת מכונה.



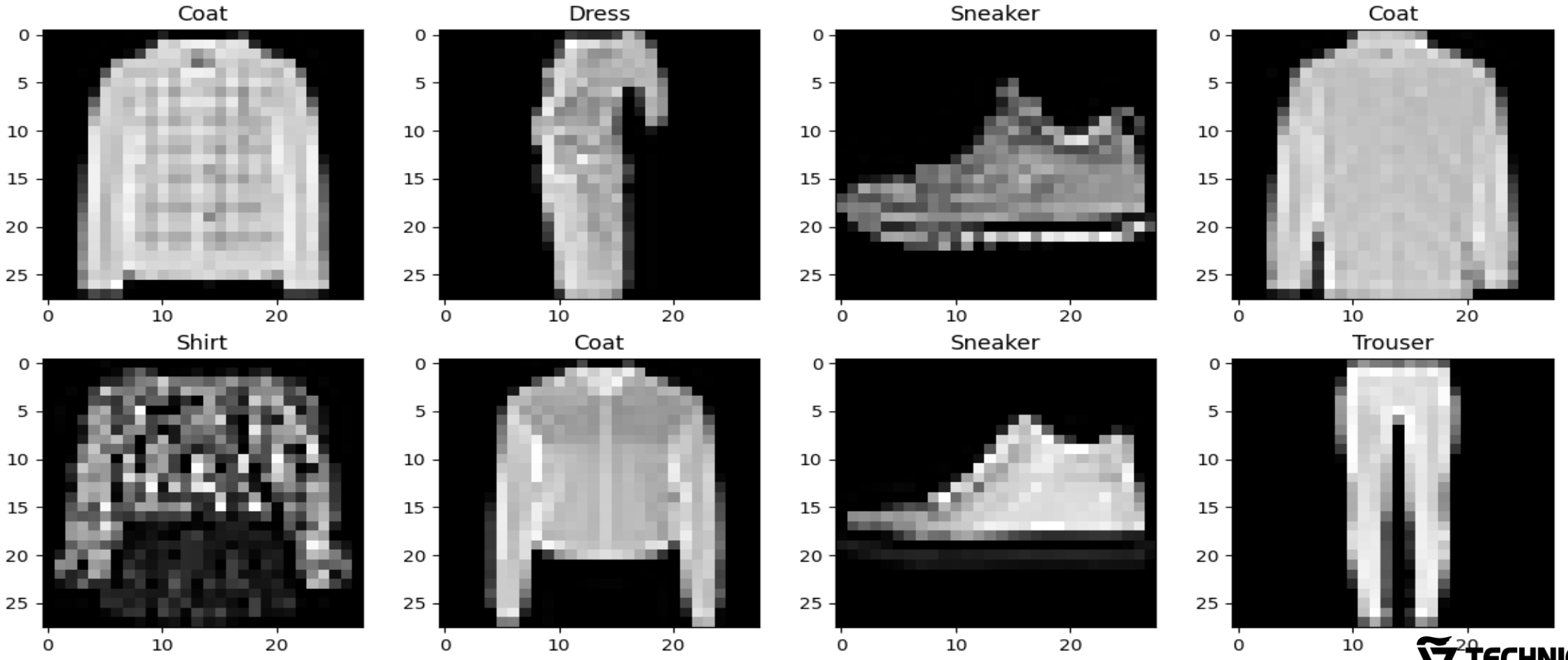
פשוט להבנה – כל תמונה היא ספרה בין 0 ל-9 בגודל **28 28xפיקסלים בגווני אפור**.

```
from tensorflow.keras.datasets import mnist

# טעינת הנתונים
(train_images, train_labels), (test_images, test_labels) = mnist.load_data()
```

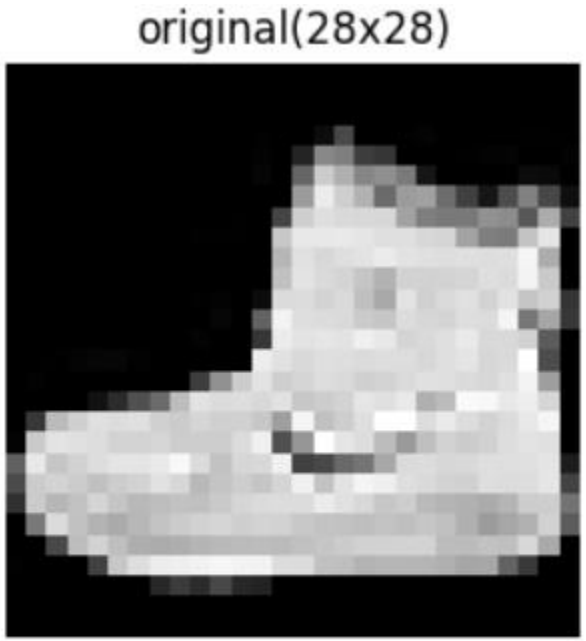
מסדי הנתונים MNIST/Fashion-MNIST

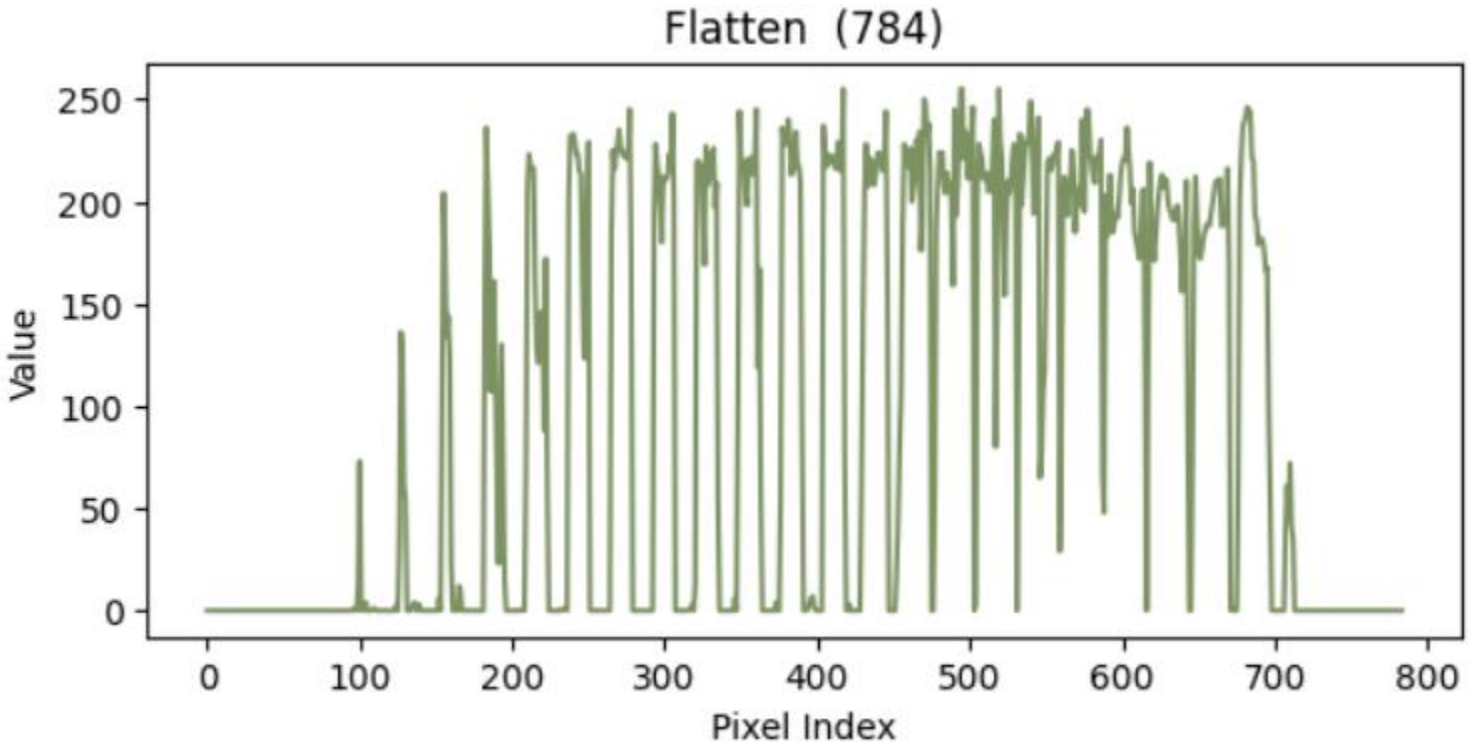
Fashion-MNIST הוא גרסה מתקדמת יותר, המכילה תמונות של בגדים ואביזרים מ-10 קטגוריות שונות, ונחשב מאתגר יותר.



שכבת Flatten

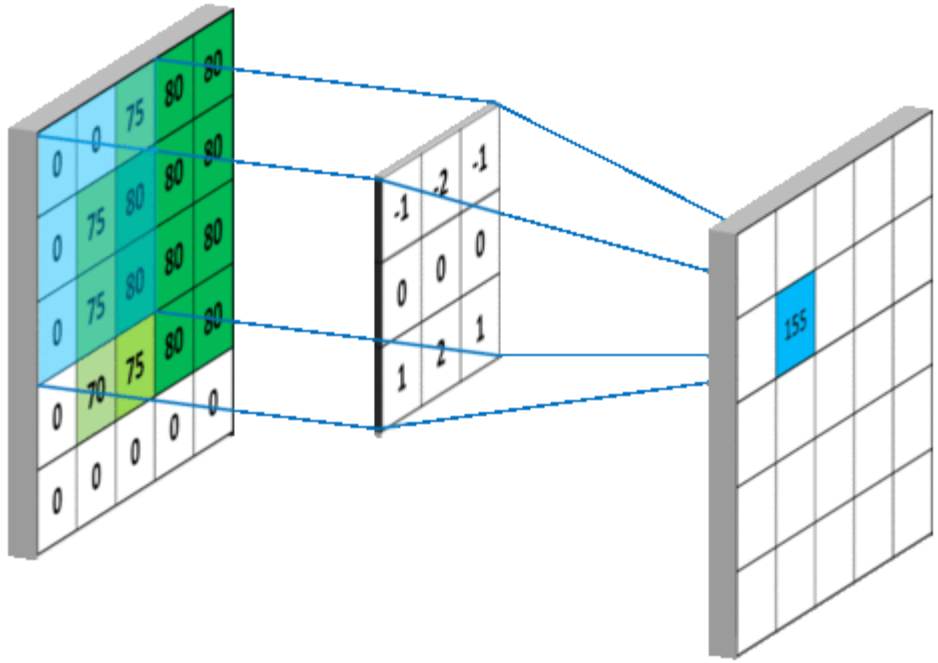
ממירה תמונה דו-ממדית לווקטור חד-ממדי, כך שניתן להזין אותה לשכבות צפופות (Dense) ברשת הנוירונים. היא אינה מבצעת חישובים, אלא רק מסדרת את הנתונים כך שיהיו מוכנים לעיבוד ע"י המודל.





שכבת Conv2D היא שכבה מרכזית ברשתות נוירונים קונבולוציוניות (CNN)

היא משמשת לחילוץ תכונות מתוך תמונה על ידי החלת פילטרים (kernals) שמבצעים קונבולוציה על האזורים המקומיים בתמונה. השכבה מזהה דפוסים כמו קווים, קצוות, ומרקמים אשר משמשים להבנת התמונה בשלבים עמוקים יותר של הרשת.



https://poloclub.github.io/cnn-explainer/#article-flatten

שכבת Conv2D

– **filters**: מספר הפילטרים (kernels)שמופעלים על התמונה. כל פילטר מחלץ סוג אחר של תכונה מהתמונה (למשל, קווים אופקיים, קווים אנכיים, מעגלים וכו'). מספר הפילטרים קובע כמה feature maps יווצרו.

– **kernel_size**: גודל הפילטר (למשל, 3×3 או 5×5). הפילטר סורק אזורים קטנים בתמונה ומחשב תכונות מתוך מידע מקומי.

– **strides** קובע בכמה פיקסלים הפילטר זז בכל צעד. ערך ברירת המחדל הוא 1, כלומר הפילטר עובר פיקסל אחרי פיקסל. אם נגדיר stride=2

```
tf.keras.layers.Conv2D(filters=10,          # 10 פילטרים שונים
                       kernel_size=(3, 3), # 3 פיקסלים x3כל פילטר בגודל
                       strides=(1, 1),     # 1 פיקסל לכל כיוון זז
                       padding='valid',    # אין Padding → התמונה תקטן
                       activation='relu',  # ReLU שכבת האקטיבציה היא
                       input_shape=(100, 100, 3))  # 100 קלט בגודל x100 ערוצים 3 עם (RGB)
```

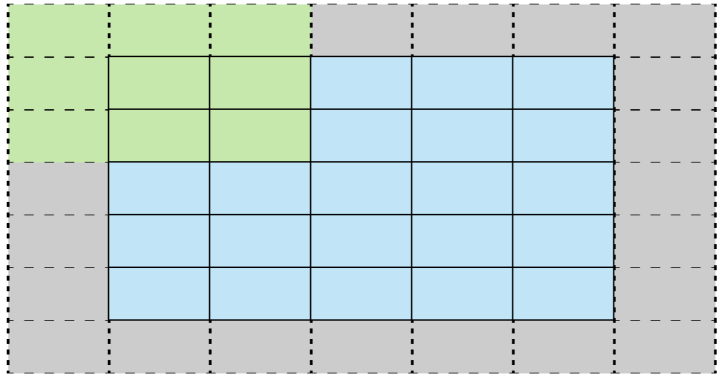
https://poloclub.github.io/cnn-explainer/#article-flatten

שכבת Conv2D

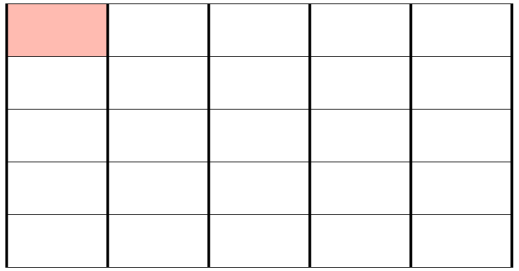
padding – קובע האם להוסיף פיקסלים מלאכותיים סביב התמונה כדי למנוע איבוד מידע בקצוות.

'padding='valid – ללא padding, מה שגורם לפלט להיות קטן יותר מהקלט.

'padding='same– מוסיף אפסים מסביב לתמונה כך שהפלט יהיה באותו גודל כמו הקלט.



Stride 1 with Padding



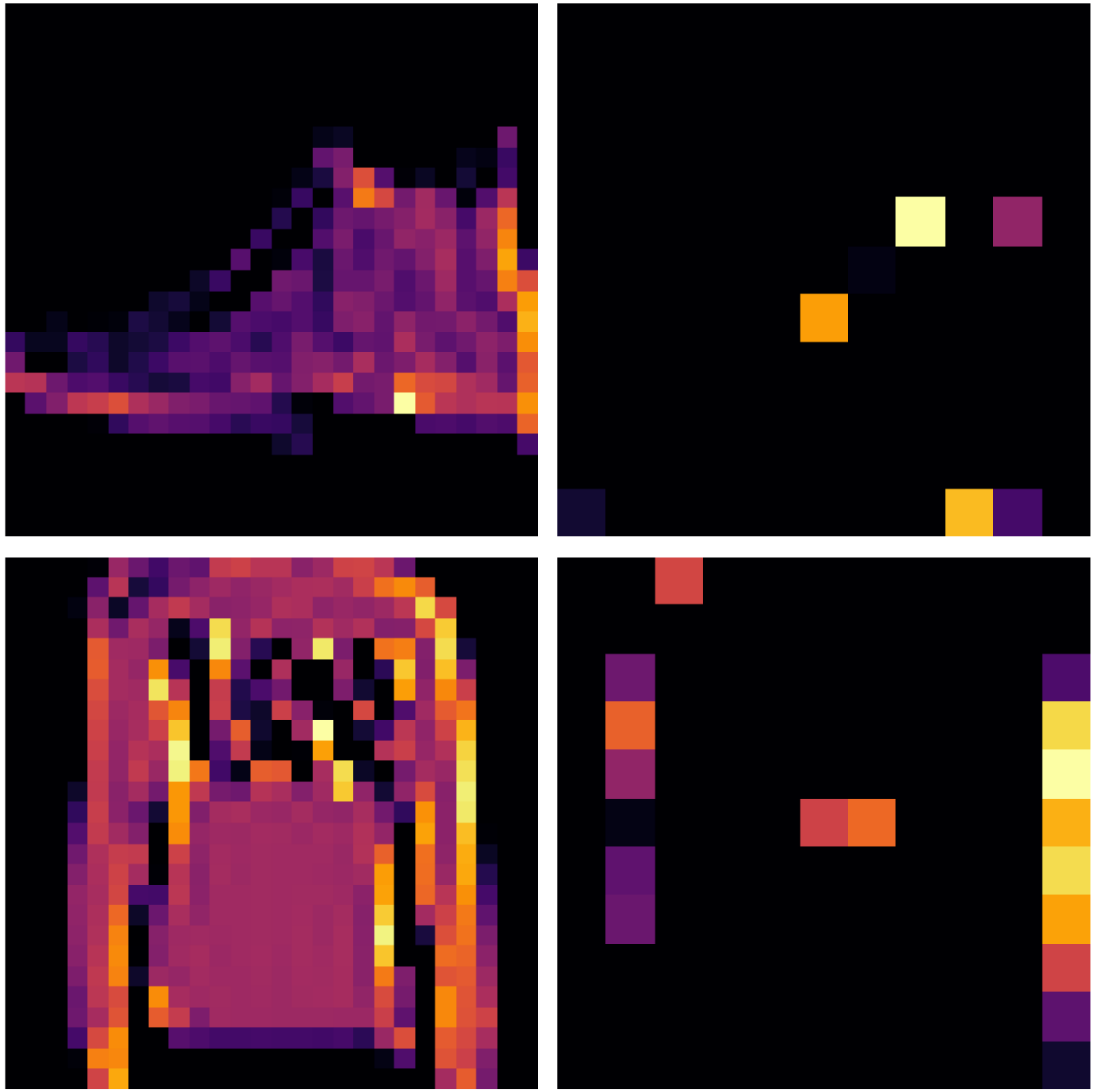
Feature Map

```
tf.keras.layers.Conv2D(filters=10,          # 10 פילטרים שונים
                       kernel_size=(3, 3), # 3 פיקסלים x3 כל פילטר בגודל
                       strides=(1, 1),     # 1 פיקסל לכל כיוון זז
                       padding='valid',    # אין Padding → התמונה תקטן
                       activation='relu',  # שכבת האקטיבציה היא ReLU
                       input_shape=(100, 100, 3))  # 100 קלט בגודל x100 ערוצים 3 עם (RGB)
```

https://poloclub.github.io/cnn-explainer/#article-flatten

שכבת Conv2D

After CNN: 28x28 pixels to 5x5 pixels

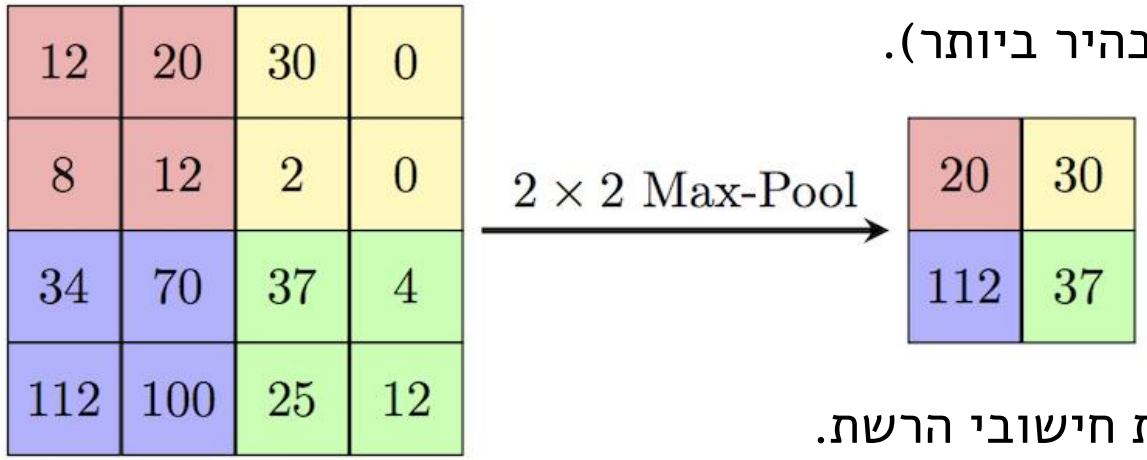


Max Pooling היא פעולה שמטרתה להקטין את ממדי התמונה תוך שמירה על התכונות החשובות שבה.

זהו שלב חשוב ברשתות CNN שמאפשר להפחית את כמות החישובים מבלי לאבד מידע קריטי.

איך זה עובד?

- מחלקים את התמונה לחלונות קטנים (למשל, בגודל 2×2 או 3×3).
- בכל חלון כזה, נשמר רק הערך הגדול ביותר (כלומר, הפיקסל הבהיר ביותר).
- ממזערים את גודל הנתונים ומבליטים תכונות חשובות.



יתרונות

- **הפחתת ממדי הנתונים** ← התמונות קטנות יותר, מה שמייעל את חישובי הרשת.
- **שמירה על התכונות החשובות** ← הערכים החזקים ביותר נשמרים.
- **עוזר בהכללת המודל** ← פחות רגיש לשינויים זניחים בתמונה.

z

שכבת Max Poolling

למשל....

גודל הפלט	פעולה	שכבה
100×100	תמונה מקורית	Input (100×100)
98×98	קונבולוציה בלי Padding	Conv2D (3×3, stride=1, padding='valid')
100×100	קונבולוציה עם Padding	Conv2D (3×3, stride=1, padding='same')
49×49	מקס-פולינג שמקטין פי 2	MaxPooling (2×2)

Image generator

Image generator

```
datagen = ImageDataGenerator(
    rotation_range=120,    # סיבוב אקראי עד 30 מעלות
    width_shift_range=0.2,  # הזזת רוחב עד 20% מגודל התמונה
    height_shift_range=0.2,  # הזזת גובה עד 20% מגודל התמונה
    shear_range=0.5,  # עיוות זוויתי
    zoom_range=0.5,  # 20% זום פנימה/החוצה עד
    horizontal_flip=True,  # היפוך אופקי
    fill_mode='nearest'  # מילוי פיקסלים חסרים
)
```

Image generator

מה זה עושה?	יתרון
מטעין תמונות **בזמן אמת** במקום לשמור הכול בזיכרון	**זיכרון יעיל**
משנה תמונות **כדי להעשיר את הדאטסט בלי לאסוף יותר תמונות**	**Data Augmentation**
מחלק את הערכים ב-255 כך שהמודל יפעל טוב יותר	**נורמליזציה אוטומטית**
קורא תמונות **ישירות מתיקיות** ומחלק אותן לקטגוריות אוטומטית	**טעינת נתונים מסודרת**

Computer Vision

- ✓ MINST Data Set
- ✓ Flatten Layer
- ✓ Convolutional Layer
- ✓ Max Pooling
- ✓ Data Augmantion