# ראייה ממוחשבת

## הקדמה

ראייה ממוחשבת (Computer Vision) היא ענף מחקר של מדעי המחשב, העוסק בעיבוד אוטומטי של תמונות המבוססות על העולם האמיתי, במטרה לחלץ ולפרש מידע חזותי הטמון בהם. נתוני התמונה יכולה ללבוש צורות רבות, כגון רצפי וידאו, תצוגות ממצלמות מרובות, או נתונים רב ממדיים מסורק רפואי.

העיניים האנושיות מתפקדות כמעין מצלמות וידאו צבעוניות ברזולוציה גבוהה ביותר והמידע הרב שנקלט מעובד מנותח במהירות שיא, תוך ניפוי הטפל מהעיקר ופיענוח המושתת על בינה אנושית. אופן פעולתו של מנגנון זה, המאפשר לתינוק בן שנתיים לזהות את אימו, או את היכולת לקרוא כל כתב יד, כמו גם את היכולת לזהות איום מכל גורם אפשרי, עדיין איננו ברור לחלוטין. חוקרים של מדעי המוח והראייה הצליחו לפענח באופן חלקי את התהליך של ניתוח הראייה אצל חיות ובני אדם. תחילת העיבוד ודחיסת המידע מתרחש כבר בשכבות השונות של הרשתית (כמו למשל חידוד הבבואה, הדגשת גבולות האובייקטים וגילוי תנועה) וחלק אחר במרכזים שונים בקליפה האחורית של המוח. הבבואה שכבר עברה עיבוד תמונה ודחיסה, יוצאת מהרשתית ברזולוציה של 1.2 מגה פיקסל באמצעות עצב הראייה, וממשיכה בעיקר אל האזור הראשוני בקליפת המוח הראייתית (V1), ומשם משוכפלת בכל אחד מאזורי ניתוח הראייה פעם נוספת, ועוברת בכל אחד מהם תהליך של ניתוח ייחודי. כך למשל נראה כי התאים באזור המכונה V1 רגישים לאוריינטציה של קווים אנכים ואופקיים וכן לתנועה בכיוון מסוים, באזור שכונה V2 התגלתה רגישות לקווים מדומים, באזור המכונה V4 התגלתה רגישות לצבעים, ואילו באזור MT התגלתה רגישות לתנועה.

ענף הראייה הממוחשבת מנסה ליצור כלים לוגיים כדי לחקות את הניתוח ועיבוד הנתונים שהמוח האנושי מבצע בעקבות הראייה, אך אף שישנן הצלחות יפות בעניין, נראה כי בהרבה תחומים המחקר עדיין בחיתוליו, וכי ידרשו עוד שנות מחקר רבות עד שיהיה אפשר להבין את מורכבותה של מערכת הראייה האנושית וחיקויה בכלים ממוחשבים. יתרון המיכון בעניין זה הוא ברור: המידע שמצלמות יכולות לקבל, לעבד ולנתח הוא לאין שיעור מדויק וכמותי מאשר העין והמוח האנושי מסוגלים לשאת, אף ברזולוציה יחסית נמוכה. כך למשל תוכנה לזיהוי טביעות אצבעות יכולה לסרוק בזמן קצר יחסית מאגר של מיליוני טביעות אצבעות כדי למצוא התאמה, סריקה שבבדיקה אנושית ידנית הייתה מבצע בלתי אפשרי.

הדרך העיקרית לניתוח התמונה נעשית על ידי פירוק התמונה לאבני הבסיס שמרכיבות אותה - פיקסלים. כך נעשה ניסיון להגדיר ולזהות דפוסים ותבניות של קבוצות פיקסלים ושינויים של צבעי הפיקסלים המתרחשים במעבר מתמונה לתמונה. לדוגמה, בגילוי תנועה מנסים לזהות שינויים בקבוצות פיקסלים מתמונה לתמונה ואף לנסות לאגד קבוצות פיקסלים קרובות לידי אובייקט אחד.

## בעיות בראייה ממוחשבת

תחום הראייה הממוחשבת מחולק ל-3 תחומים:

### מדידת הסביבה

* 3D Reconstruction - הדמיית תלת ממד. הפקת תמונה תלת ממדית ממצלמה או מזוג מצלמות. לכל פיקסל בתמונות המקוריות יש שני ממדים (x, y), המטרה היא להמיר כל פיקסל כזה לשלושה ממדים (x, y, z), כלומר להוסיף לכל פיקסל את ממד העומק. ישנן מספר דרכים שניתן ליישם זאת:
* צילום סטריאוסקופי, שבו זוג מצלמות או יותר מצלמות את אותו אזור מזווית קצת שונה, וניתוח ההבדלים על מנת להפיק את ממד העומק בדומה לתהליך שעושות העיניים.
* ניתוח האור וההצללה של אובייקטים ובאמצעותם חישובי עומק של העצמים.
* באמצעות חישובי התנועה של העצמים.
* יצירת מפת עומק באמצעות הקרנת אור אינפרה-אדום וקליטתו באמצעות חיישן שמבצע את חישוב העומק.
* מדידת עצמים - למדוד את צורתם, מיקומם וצבעיהם של עצמים. חישוב של פרספקטיבה והערכת מרחק של עצמים מהמצלמה. למשל מערכות רובוטיות הבונות מכונית, שצריכות לאתר אזורים מסוימים כדי לבצע בהן פעילות מסוימת. או מערכות רמזור המודדות את מהירותם של מכוניות המגיעות לרמזור.
* ניווט אוטומטי - בתחום זה מנסים להקנות לרובוט או המכונית, אפשרות לפענח את השטח ולנוע בו באופן חופשי, באופן שלא יתקל בעצמים או במכשולים.

### ניתוח תנועה

זהו תחום שבו התוכנה מזהה ומבודדת תנועה של עצמים, באמצעות השוואת השינויים שבין פריימים של תמונות הווידאו. ענף זה מנסה לחקות את יכולת האדם ובעלי חיים נוספים להבחין בתנועת עצמים במרחב. יש לסנן רעשים, שינויי תאורה, ותנועות של עצמים לא רצויים, כדי לאבחן תנועה של העצמים המבוקשים. תחום זה הוא אחד הפופולריים בראייה הממוחשבת, ונעשה בו שימוש למטרות ניטור, השגחה, אבטחה, מדידה, ניתוח דפוסים ועוד.

* עקיבת וידאו (Video tracking) - מערכת המאפשרת עקיבת המצלמה אחרי האובייקט. בתחום זה יש להניע את המצלמה בעקבות האובייקט, וצריך לוודא שהעיקוב מתבצע אחרי האובייקט שנבחר, כך שכאשר הוא עובר מאחורי עצמים אחרים, או כאשר הוא נפגש בעצמים אחרים התוכנה עדיין מתבייתת עליו, ולא טועה לעבור לאובייקט אחר.
* בקרת תנועות חריגות - מערכת הלומדת תנועות שגרתיות של עצמים במרחב מסוים ומסוגלת לדווח כאשר התנועות של העצמים במרחב חריגות ויוצאות דופן. למשל בקרת תנועה חריגה של אנשים ורכבים בצומת, או כאשר רכבים יורדים מהכביש, בניגוד לתנועה השגרתית שבה כלי רכב נעים על הכביש.

### זיהוי והכרה

הבעיה הקלאסית של ראייה ממוחשבת היא לקבוע אם התמונה מכילה אובייקט, תכונה מסוימת, או פעילות. משימה זו שבדרך כלל יכול להיפתר ללא מאמץ בידי אדם, עדיין לא נפתרה באופן משביע רצון במקרים כלליים. בשיטות הקיימות להתמודדות עם בעיה זו ניתן לפתור אותה, במקרה הטוב, רק עבור אובייקטים ספציפיים, כגון אובייקטים בעלי גאומטריה פשוטה (למשל פּוֹלִיהֶדְרוֹן), פרצופים אנושיים, כתב יד מודפס, דמויות או כלי רכב, ובמצבים מסוימים, כמו תנאי תאורה המוגדרים היטב, רקע מסוים, ועמדה של האובייקט ביחס למצלמה.

* Identification - לזהות מה מופיע בתמונה. לאחר שהמערכת מזהה שאובייקט מסוים הוא אובייקט שלם, היא יודעת להגדיר את טבעו, בהשוואה לתבניות מוכנות מראש או באמצעות השוואה לווקטור מאפיינים של תבניות קיימות. בדרך כלל מדובר במספר טיפוסי אובייקטים קטן, המוגדר מראש בתחום מסוים. זיהוי מסוג זה נקרא Class based. תחת תחום זה נמצא סיווג אובייקטים (Object Classification).
* Detection - לגלות האם בתמונה יש אובייקט ספציפי והיכן נמצא או האם התמונה מקיימת תכונה כלשהי. זיהוי מסוג זה נקרא Instance based.
* Segmentation - הפרדה בין האובייקט לרקע, ניתוח בולטות וחשיבות של אובייקטים בתמונה או בווידאו. נעשה בין היתר באמצעות איתור דפוסים ייחודיים בתמונה. למידה עמוקה הקפיצה תחום זה.
* פענוח כתב אופטי (Optical Character Recognition - OCR) - "הבנת" הכתב האופטי והמרתו לכתב מוקלד. תהליך שהאדם עושה בעיניו ומוחו. ההפך מהדפסה. דרך פשוטה היא באמצעות התאמה מבנית, שבה נעשית השוואה בין האות לבין דוגמאות של אותיות הנמצאות במאגר. מוגבלותה של שיטה זו בכך שהיא מוגבלת רק לצורות הקיימות במאגר. דרך שנייה היא באמצעות תיאור מבני שבה כל אות מתוארת באמצעות הקווים המרכיבים אותה והיחסים שביניהן. שיטה זו שמורכבת ליישום מקודמתה, היא גם כוללנית, גמישה ומוצלחת יותר. תת תחום של OCR הוא זיהוי אוטומטי של לוחיות רכב.
* Event classification - הבנת פעילותם והתנהגותם של אובייקטים במרחב מסוים. כך למשל נעשה ניסיון להבין באמצעי ממוחשב כאשר מתרחשת אינטראקציה אלימה בין בני אדם, או כאשר מתרחשת פעילות פלילית.
* זיהוי תוואי שטח וקריאת מפות - דומה ל-OCR רק שהקריאה כאן היא של תמונת שטח והבנת הטופוגרפיה שלה.
* זיהוי ביומטרי חזותי - בדיקת התאמה בין בבואה חזותית של אובייקט פיזי לבין תמונות במאגר כמו פנים, רשתית, וטביעות אצבעות. בתחום של זיהוי פנים (Face recognition), צורת הפנים, גודל מרכיבי הפנים והיחס ביניהם ממוצים לנתונים שנשמרים במאגר (ווקטור שינויים), כמו למשל המרחק בין העיניים, המרחק בין האוזניים, גודל הפה יחסית לפנים ומרחקו מהאף וכדומה. כאשר מקבלים פנים חדשות שצריך לאתר הופכים אותם לווקטור שינויים ומחפשים התאמה במאגר. כמובן שהאלגוריתם הרבה יותר מורכב, מכיוון שהוא צריך לזהות פנים מכל מיני זוויות, ויכולים להיות שינויים בהם כמו חבישת כובע, משקפיים או גידול שפם.

## קשיים בניתוח תמונה

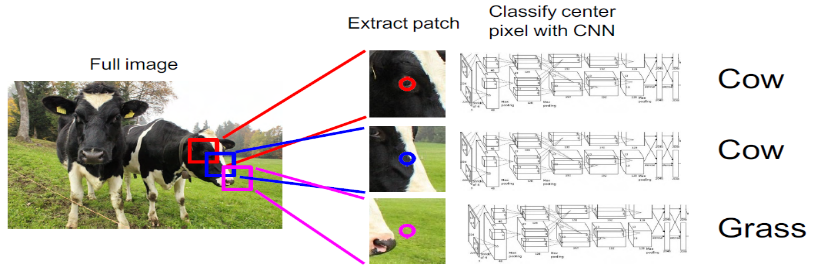
* שינוי בזווית מיקום המצלמה ביחס לאובייקט.
* רקע דומה לאובייקט.
* תאורה מעומעמת.
* רק חלק מהאובייקט נראה בתמונה.
* האובייקט מוצג בצורה שאינה טבעית לו או מעוותת
* יש מספר אובייקטים ולא רק אחד.

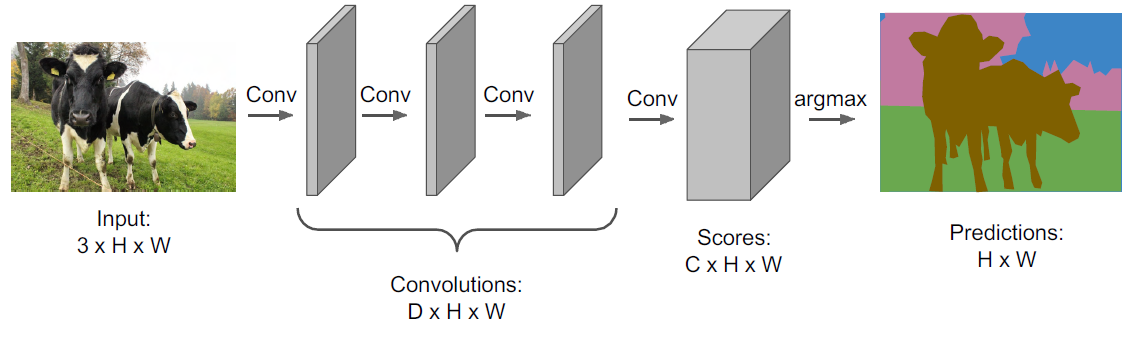
# Detection and Segmentation

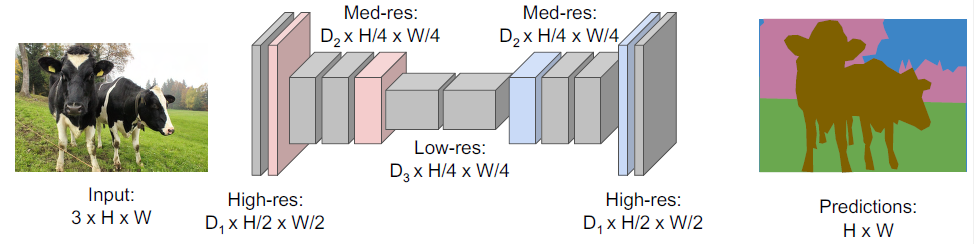
## Semantic Segmentation

זוהי בעיה שבה בהינתן תמונה נרצה לסווג כל פיקסל בתמונה לאיזה מחלקה הוא שייך. הדאטה צריך להיות זוגות של תמונה ו-label לכל פיקסל בתמונה שמציין לאיזה מחלקה שייך הפיקסל.

דרך אחת לעשות זאת הוא לחלק את התמונה לריבועים קטנים (patches), ועבור כל ריבוע להפעיל אלגוריתם CNN כדי לסווג לאיזה מחלקה ה-patch הזה שייך. כל הפיקסלים ב-patch יקבלו אותו סיווג. בשלב האימון נחלק כל תמונה ב-train ל-patches כאשר ה-label לכל תמונה הוא הפיקסל האמצעי. יש שני בעיות בשיטה זו:

1. ב-train יהיו לנו המון patches כך שתהליך הלמידה יהיה ארוך ביותר עד לא אפשרי.
2. ניסיון לסווג patch קטן ויחיד ללא ההקשר (הפרטים מסביבו) סביר להניח שיניב תוצאות לא טובות.

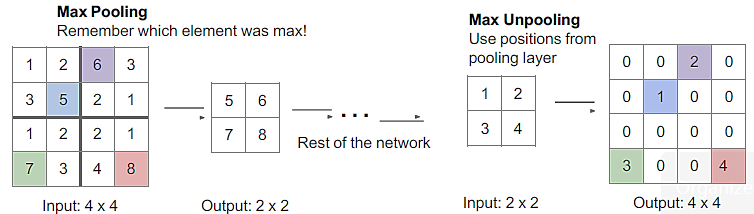
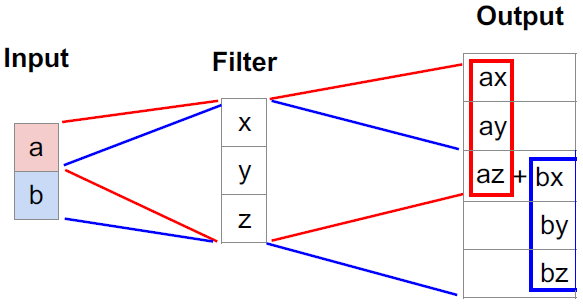
דרך נוספת שתפתור בעיות אלו, להשתמש באלגוריתם CNN על התמונה כולה, כך שהרשת מורכבת רק משכבות קונבולוציה שאינן מקטינות את המימד. הפלט יהיה בגודל התמונה המקורית ויציין לכל פיקסל לאיזה מחלקה שייך. לשיטה זו אמנם תהיה בעיה חדשה שהרשת תכלול המון פרמטרים ושוב נקבל תהליך למידה מאוד ארוך ולא אפשרי.

הפתרון הוא אכן להשתמש ב-CNN על התמונה כולה באמצעות שכבות קונבולוציה בלבד, אלא שבהתחלה נצמצם את המידע (downsampling) ולבסוף נרחיב אותו עד שיגיע לגודל התמונה המקורית (upsampling). הקושי הוא לעשות זאת בלי לאבד מידע.

תהליך ה-downsampling הוא פשוט באמצעות הגדלת ה-stride, פילטרים ו-pooling. תהליך ה-upsampling לעומת זאת הוא מושג חדש אותו נלמד עתה לראשונה.

### upsampling

יש מספר שיטות כיצד לבצע upsampling:

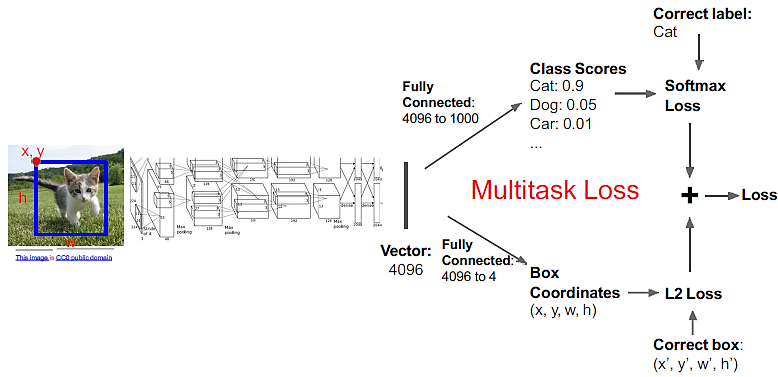
1. Max Unpooling – בכל פעם שמבצעים Max Pooling בנוסף לשמירת הפיקסל המקסימלי שומרים גם את המיקום שלו. כאשר רוצים להרחיב את המידע, נוסיף שכבה הפוכה ל-Max Pooling שמרחיבה כל פיקסל לריבוע שכולו אפסים מלבד המיקום של האיבר המקסימלי שבו יהיה ערכו של הפיקסל.
2. Transpose Convolution - לפי שיטה זו נבחר מהו גודל המידע שנרצה שיהיה ב-output לשכבה הבאה ואז "נלמד" מהי קבוצת הפילטרים שתשנה מידע ב-input של שכבה נוכחית לגודל המידע הרצוי בשכבה הבאה. אנו מנסים למצוא מהי קבוצת הפילטרים שאם נפעיל אותם על ה-output ייתנו לנו את ה-input, זוהי בעצם קונבולוציה הפוכה. לשם כך נבנה משוואות לכל ערך ב-input של איזה ערכים צריכים להיות בפילטרים כדי להגיע לערך זה. האלגוריתם יילמד מהם הערכים הטובים ביותר לפילטרים. נשתמש בפילטרים כדי ליצור את ה-output בשכבה הבאה. **זוהי השיטה המומלצת לביצוע upsampling**, כיוון שגם לומדת מהם הערכים האופטימליים.

## Object Detection

זוהי בעיה שבה בהינתן תמונה נרצה למצוא את כל האובייקטים שנמצאים בה ולהקיף כל אובייקט ב-bounding box, התוחמת את גבולות האובייקט. במידה ויש שני אובייקטים זהים נרצה שכל אחד מהם יהיה ב-bounding box שונה, כלומר נרצה גם לזהות ולהפריד מספר אובייקטים מכל סוג.

בדאטה לבעיה זו, לכל אובייקט בתמונה צריך להיות נקודה (x, y) המציינת את הגבול השמאלי העליון של ה-bounding box, גובה h ורוחב w של ה-bounding box, ו-label לאיזה מחלקה שייך האובייקט. סה"כ 5 תכונות נוספות לכל אובייקט. הפלט לבעיה הוא bounding box לכל אובייקט ורשימת של ההסתברויות ששייך לכל מחלקה שאנו מסווגים אליה. נשים לב שלכל תמונה יש גודל פלט שונה בהתאם למספר האובייקטים המופיעים בה.

### תמונה עם אובייקט יחיד

עבור תמונה עם אובייקט יחיד, נוכל לבנות רשת CNN שבשכבות האחרונות שלה ניצור שכבה שתחזה את הסתברות השייכות של האובייקט לכל מחלקה באמצעות Softmax, ושכבה מקבילה שבה תחזה מהו ה-bounding box בתוך התמונה בו נמצא האובייקט. פונקציית ה-loss מורכבת מחיבור של Softmax loss ומרחק בין הנקודות של ה-bounding box האמיתי לחיזוי.

### R-CNN

זהו אלגוריתם שיכול לזהות מספר אובייקטים בכל תמונה. מספר האובייקטים כמובן יכול להשתנות בהתאם לתמונה שמקבל. החידוש באלגוריתם זה הוא שמשתמש באלגוריתם ROI שיודע לזהות את החלקים החשובים בכל תמונה שבהם יכולים להיות אובייקטים. ROI משתמש במספר שיטות כמו edge detection ובממוצע מוצא 2000 חלקים חשובים בכל תמונה. כל חלק מנרמלים כדי שיהיה גודל אחיד ומכניסים לרשת CNN, כמו שהסברנו לעיל, שתבדוק אם יש אובייקט בחלק זה ותמצא לו bounding box. לבסוף, בוחרים את כל החלקים עם ההסתברויות הגדולות ביותר שאינן חופפות. הבעיה באלגוריתם זה הוא שהוא איטי מאוד ולא ניתן להשתמש בו במערכות מהירות. הומצאו מספר אלגוריתמים שמייעלים אותו.

### Fast R-CNN

השיפור העיקרי שקודם מכניס את כל התמונה לרשת CNN כדי לכווץ אותה, ועל הפלט מפעיל את אלגוריתם ROI.

### Faster R-CNN

משפר עוד יותר בכך שמחליף את אלגוריתם ROI ברשת שיודעת במקביל גם לחשב loss.

# עיבוד תמונה

## ייצוג של תמונה

עיבוד תמונה (Image Processing) הוא תחום במדעי המחשב העוסק בעיבוד של תמונות ובמניפולציות שונות עליה. כל קובץ תמונה הוא מטריצה של נקודות זעירות הקרויות פיקסלים. ניתן גם להסתכל על תמונה כפונקציה , שבה כל נקודה היא פיקסל. לכל תמונה יש שני פרמטרים:

1. רזולוציה – גודל המטריצה. ככל שיש יותר נקודות ביחידת שטח, כך הרזולוציה של התמונה גבוהה יותר, והתמונה תראה חדה יותר. הפחתת רזולוציה פי x אומרת לחלק את התמונה לריבועים, שבכל אחד מהם יש x פיקסלים, ולהמיר את כל הפיקסלים בערך אחד (בדרך כלל ממוצע).
2. Quantization – מהו הערך המקסימלי בכל פיקסל. הגודל הסטנדרטי הוא 255. ככל שפיקסל יכול לתת טווח יותר גדול של צבעים, כך עומק הצבע גדול יותר, והתמונה תראה ריאליסטית יותר. בנוסף, ככל שפיקסל בהיר יותר כך הערך שלו גדול יותר.

בתמונה עם צבעי אפור בלבד (Gray Level), התמונה היא מטריצה בשני ממדים . כל פיקסל מורכב ממספר אחד המייצג את בהירות האפור, כאשר 0 זה שחור ו-255 זה לבן. בתמונה עם צבע, התמונה היא מטריצה בשלושה ממדים . כל פיקסל מורכב משלושה מספרים (r, b, g).

## עיבוד תמונה

עיבוד תמונה נעשה לשתי מטרות עיקריות:

1. לשם שיפור התמונה: הפחתת "רעשים", הדגשת פרטים, חידוד, ריטוש תמונה, יצירת אפקטים, תיקון צבעים ורמות אפור וכדומה. זו המטרה העיקרית בתחום זה.
2. לשם הסקת מסקנות: מדידות, זיהוי וניקוי פגמים, גילוי תנועה, חיפוש צורות ואובייקטים.

לעיבוד תמונה שימושים רבים:

* בתעשייה עיבוד התמונה משמש למדידות ולזיהוי פגמים. ישנן גם מכונות ש"רואות" את העובד המשתמש בהן, מכוונות את כלי העבודה ואחר בודקות את התוצאה.
* בביטחון לאומי, בביטחון פנים ובאבטחה עיבוד התמונה משמש לזיהוי תנועה וסיווג צורות.
* בציבור הרחב משתמשים בעיבוד תמונה על מנת לשפר את התמונות שצולמו.

## סוגי פעולות עיבוד תמונה

נציין מספר סוגי פעולות פשוטות על תמונות.

### Point-based

מפעילים פונקציה לינארית על כל פיקסל. פעולות אלו תלויות אך ורק על גודל הפיקסל ולא במיקומו (position free) וכן לא בפיקסלים שלידו (context free).

### Intensity Transformations

כמו הקודם אלא שהפונקציה יכולה להיות לא לינארית. לפעולה מסוג זה יכולה להיות השפיה גדולה ושימושית על התמונה. נביא מסר דוגמאות:

1. ניגודיות – החלפת כל פיקסל בהופכי שלו. .
2. Gamma Correction – העלאת כל פיקסל בחזקה קבועה. . עבור פעולה זו תגרום לפיקסלים להיות כהים יותר באופן יחסי. כיוון שהעין שלנו יותר רגישה לשינויים בצבעים כהים פעולה זו תשפר את איכות התמונה.

### Geometric operations

מפעילים פונקציה על כל פיקסל שתלויה במיקום של הפיקסל בתמונה ולא בערך של הפיקסל. פעולות אלו משמשות לסיבוב, היפוך ומראה של התמונה.

### Spatial operations

מפעילים פונקציה על כל פיקסל שתלויה בערך של הפיקסל, במיקום שלו ובשכנים שלו. פעולות אלו מאוד שימושיות ומשמשות בעיקר להפעלת פילטרים על התמונה.

## Filtering

זוהי שיטה שבה יוצרים מתמונה קיימת תמונה חדשה כתוצאה מ-Spatial operation על התמונה המקורית. בפעולות אלו עבור כל פיקסל (x,y) בתמונה המקורית, לוקחים חלון בגודל מסביבו ומפעילים פונקציה, הנקראת פילטר, על ערכי הפיקסלים בחלון ומחזירה ערך אחד שיכנס במיקום (x,y) בתמונה החדשה. פילטרים מאוד שימושיים בעיבוד תמונה. השימושים המרכזיים שלהם הם: ניקוי רעש, חיפוש תבניות, וזיהוי קצוות.

פילטרים שימושיים:

* Salt & pepper noise - הוספת נקודות לבנות ושחורות לתמונה באקראיות על ידי הפונקציה משמאל.
* Median filter – משמש לניקוי רעש בתמונה. מכל חלון לוקח את הערך האמצעי.
* Average filter – משמש לניקוי רעש ועמעום של התמונה. מכל חלון מחזיר את הממוצע.

## Template Matching

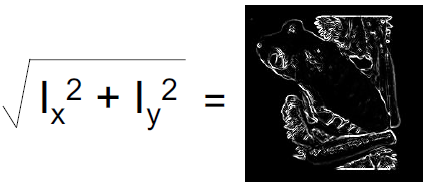
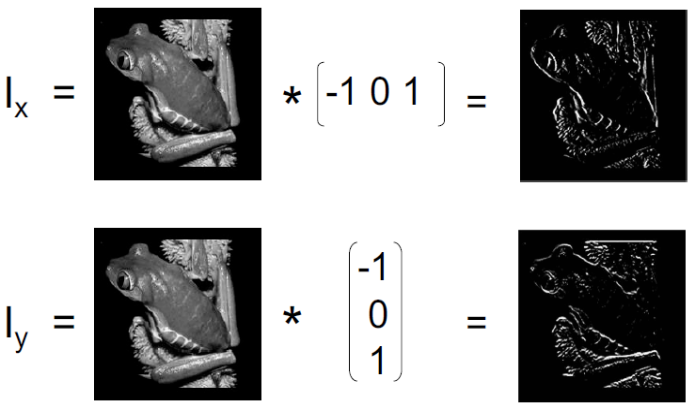
בהינתן תמונה ותבנית נרצה למצוא מיקום בתמונה שבו נמצאת התבנית. האתגר המרכזי כאן הוא שהתבנית אינה מופיעה בתמונה בדיוק כמו המקור אלא עם רעש או בהירות שונה. הדרך לפתור את הבעיה הוא שעבור תבנית בגודל נעבור על כל חלק בתמונה בגודל ונפעיל עליו פילטר. נחזיר את הפיקסל האמצעי בחלק עם הערך הגדול ביותר.

הפילטר הכי יעיל לצורך מטרה זו הוא Zero normalized cross correlation. קודם מחסירים מהתבנית וכל חלק שמפעילים עליו את הפילטר את הממוצע, כדי לתקן שינויים בבהירות. לאחר מכן ננרמל את התבנית והחלק בתמונה כדי להוריד רעש. לבסוף נכפיל מכפלה וקטורית בין התבנית והתמונה.

## Edge Detection

*בהינתן תמונה נרצה ליצור תמונה חדשה שתציג את כל הקצוות של התמונה המקורית. קצה הוא שינוי חד בגוון של פיקסל. המוטיבציה למציאת קצוות היא שבאמצעותם ניתן לזהות אובייקטים ולהפיק מידע מאוד משמעותי על התמונה.*

*כדי למצוא קצוות נסתכל על התמונה כאל פונקציה ואז בכל נקודה נחשב את עוצמת הגרדיאנט (*Magnitude*), שייתן לנו את השינוי המקסימלי בגוון באותה נקודה, ונייצר תמונה חדשה עם כל עוצמות אלו. ניתן לעשות זאת בדרך פשוטה על ידי כך שנחשב באמצעות פילטרים את השינוי בין פיקסלים אנכיים (נגזרת לפי שורות) ובין פיקסלים אופקיים (נגזרת לפי עמודות), ונחשב באמצעותם את עוצמת הגרדיאנט.*



### Edge Detection with noise

רעש בתמונה מגדיל מאוד את הנגזרת. לכן בתמונה עם רעש טכניקת זיהוי הקצוות שתיארנו תניב המון קצוות כתוצאה מרעש ולא קצוות אמיתיים. הדרך להתמודד עם רעש בזיהוי קצוות הוא קודם לטשטש (blurring) מעט את התמונה כדי להחליק את הרעש ולאחר מכן להפעיל עליה את הטכניקה לעיל. כדי לטשטש תמונה יש להפעיל עליה פילטר גאוסיאני.

# Visualization

## הגדרה

הבנה כיצד הרשת עובדת ושימוש במידע זה.