RCNN, Fast RCNN And Faster RCNN

בניית מודלים לזיהוי אובייקטים בתמונות הוא תחום מחקרי חשוב בעולם הבינה המלכותית והראייה הממוחשבת. במסגרת זו, פותחו מספר מודלים שמטרתם לזהות את האובייקטים המופיעים בתמונה ואיכן הם נמצאים.

יש שתי גישות מרכזיות לזיהוי האובייקטים:

- region-based object detection – קודם נחפש מקומות שבהם יכולים להיות אובייקטים ואז ננסה לדייק יותר במיקום ולזהות איזה אובייקטים (שזה השיטות שנדבר עליהם בסיכום הזה).
- single shot object detection – ננסה לזהות את האובייקטים ואת המיקום בניסיון אחד ע"י חילוק של התמונה לתת תמונות, מלבנים קטנים יותר וחיפוש בכל אחד מהם.

אחת הגישות הנאיביות והמוקדמות לבעיה היא לבדוק כל מקום אפשרי שבצורת מלבן בתמונה ולבדוק האם נמצא בוא אובייקט או רקע. ז"א לתמונה בגודל HxW ישנם:
$\frac{H(H+1)W(W+1)}{4} = \sum_{h=1}^{H}\sum_{w=1}^{W}(W-w+1)(H-h+1)$ מלבנים לבדוק הבעיה בכך שבהינתן תמונה בגודל 224X224 ישנם כ 635 מיליון מלבנים לבדוק, לכן שיטה זו לוקחת המון זמן ובמיוחד לא רלוונטית למקרים שנרצה לבדוק תמונות בזמן אמת.

לכל תמונה נחשב את ה loss ע"י מציאת ההבדלים בריבוע (MSE) בין המיקום המדויק של המלבן של האובייקט לבין המלבן שמצאנו, ומציאת ה cross entropy loss לאיזה אובייקט נקלט ולאחר מכן נמצא את ה- loss הסופי כמשוואה של הקודמים.

השיטות הבאות באות לנסות לפתור בעיה זו ולעשות זאת בצורה יותר יעילה ובכך מהירה יותר.

RCNN

בדומה לגישה הקודמת קודם נרצה למצוא את המיקום של האובייקטים ואז לכל מיקום לבדוק איזה אובייקט יש שם (או שאין). ב RCNN נעזר ב selective search – אלגוריתם למציאת מקומות "מעניינים" שיש סיכוי יותר גדול שמצא בהם אובייקטים, אלגוריתם זה לא לומד ומוצא מספר גדול של מקומות שצריך לבדוק הוא מוצא אובייקטים ע"י מציאת אזורים דומים בתמונה לפי צבעים וגדלים. כל מקום אחרי זה עובד ברשת קונבולוציה ולאחר התוצאה מפוצלת לשניים:

- SVG למציאת איזה אובייקט נמצא שם (או אם לא נמצא שם).
- Regressor למציאת המיקום ה"מתוקן" של האובייקט.

הבעיה בשיטה זו היא ש selective search מוצא הרבה מקומות (או קצת עם דיוק נמוך) וכל מקום נבדק בנפרד ולכן שיטה עדיין מאוד איטית, אבל אפשר לשים לב שלוקח בערך פי 10 יותר זמן גם למצוא את המקומות וגם למצוא את האובייקטים מאשר רק למצוא את האובייקטים, וזה בעיקר בגלל ש selective search לוקח את רוב הזמן בעיבוד כל תמונה.

Fast RCNN

פתרון אחר הוא שקודם כל נריץ את התמונה ברשת קונבולוציה ורק לאחר מכן נמצא אזורים פוטנציאלים בעזרת selection search, לאחר מכן כל אזור עובר בשכבת ROI pooling ומועבר לשכבת fully connected. מאחר ואנחנו מריצים את רשת הקונבולוציה פעם אחת שיטה זו הרבה יותר מהירה.

ROI Pooling

כדי לקבל מפת פיצ'רים בגודל קבוע מתוך אזורים "מעניינים" (כמו שהוסבר מקודם ב selective search) נוכל להשתמש ב ROI Pooling.

איך זה עובד: בהינתן ROI Pooling של $h\ x\ w$ קלט תמונה בגודל $H\ x\ W$ נחלק בקווים את התמונה למלבנים (תמונות בגודל) של $\frac{h}{H} x \frac{w}{W}$ ולאחר מכן נמצא נעשה על כל מלבן Max Pooling – נחליף את הערך המקסימאלי של כל מלבן באותו מלבן. נקבל תמונה יותר קטנה שנשמרו בה הפיקסלים המקסימליים (או מטריצה שנשמרו ממנה האיברים המקסימליים אבל גודלה הצטמצם).

Faster RCNN

למרות ש Fast RCNN הרבה יותר מהיר מ RCNN, selective search הוא אלגוריתם איטי יחסית. שיטת Faster RCNN באה לשפר את הזמן ע"י החלפת האלגוריתם ב- RPN (Regional Proposal Network).

ב Faster RCNN התמונה מועברת לרשת קונבולוציה כמו בשיטה הקודמת, אך לאחר מכן מומרת ל Feature map ואז מועברת לרשת לומדת שעוברת על כל המיקומים בתמונה ומוציאה פרדיקציה האם יש שם אובייקט, ומוציאה את המלבן הרלוונטי. RPN עובד עם מלבנים שהדלים שלהם מוגדרים מראש (Anchors) והרשת מתקנת אותם לגדלים הרלוונטיים. לאחר מכן התוצאה מועברת ל ROI pooling ואז ל classifier (שמזהה את האובייקט).

Faster RCNN יותר מהיר מ Fast RCNN מאחר ואינו נעזר ב selective search.