

# עבודת סיקור מאמר - מבוא לעיבוד ספרתי של תמונות

Introduction to digital image processing.

שם המגיש: תום לב-רון.

תאריך הגשה: 2.2.2021.

שם המאמר: Scale-recurrent Network for Deep Image Deblurring.

כותבי המאמר: Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, Jiaya Jia

The Chinese University of Hong Kong, YouTu Lab, Tencent 3Megvii Inc.

## Contents

1	מבוא
1	שיטות
2	שיטות קיימות
2	השיטה המוצעת במאמר
4	תוצאות
5	דיון
5	סיכום
6	נספחים
6	Convolutional Neural Networks
6	Encoder-Decoder
6	Recurrent Neural Network
6	Long-Short-Term-Memory
7	ResBlocks
8	דוגמאות לתוצאות המחקר
10	מקורות

## מבוא

טשטוש תמונה הוא אחד מתוצרי הלוואי הנפוצים כאשר מצלמים אובייקט, בפרט אובייקט בתנועה או פתיחת צמצם בזמן ארוך. תנועות חדות של האובייקט ו/או המצלמה יכולים להוריד את איכות התמונה ולגרום לטשטוש. בנוסף, גורמים נוספים כמו עומק תמונה משתנה שלא נרכש בעומק השדה הרצוי וכדומה יכולים להערים קשיים נוספים. מחקרים נפוצים מתעסקים בעיקר בטשטושי תזוזות המצלמה, כיום מחקרים מתמקדים בסוגים שונים של טשטושים הנובעים מטשטוש לא אחיד. שיטות תיקון מקובלות ונפוצות מניחות ידע מוקדם על פונקציית המריחה או ניסיון לשערך אותו, שיטות אלו מגבילות מאוד את מאפייני שערך הטשטוש. בעיה נוספת היא שלעיתים החישובים לא מוציאים תוצאות טובות על מידע שאינו מהמעבדה, כלומר מידע "בעולם האמיתי". הסיבה לזה היא שבעולם "האמיתי" הטשטוש לעיתים יותר מסובך, מגוון ולעיתים מושפע מחישובים של חומרת המצלמה. המאמר שאסביר מנסה להציג גישה מעט שונה וינסה להיות כללי מספיק כדי לנסות ולתקן תמונות מטושטשות מהחיים ה"אמיתיים" ולא רק תמונות מתוך מאגר מידע. בסוף הקובץ הוספתי כמה נספחים על נושאים בהם הייתה נדרשת מעט הרחבה.

## שיטות

הבעיה איתה מנסים החוקרים להתמודד היא בעיה מוכרת אשר יכולה להיפתר במגוון שיטות. ישנם השיטות "הקלאסיות", אשר משתמשות בפילטרים ושערך פונקציית הטשטוש. ישנם שיטות בהם משערכים את פונקציית

הטשטוש על ידי למידה ולאחר מכן מסננים באמצעותה בעזרת פילטרים מוכרים. בשנים האחרונות ישנו דגש גדול על שימוש בשיטות למידה להסרת טשטוש מתמונות בצורה עיוורת (ללא ידע מוקדם על התמונה). החוקרים מציגים שיטה לשחזור תמונות מטשטוש כללי בעזרת רשת נוירונים אשר מקבלת רצף של תמונות ומחזירה רצף של תמונות "נקיות" מטשטוש.

## שיטות קיימות

ההתמקדות הנפוצה כיום במחקרים היא בשחזור תמונות ללא ידיעה של פונקציית המריחה, שיטה זו מכונה "תיקון טשטוש עיוור". שיטות למידה שונות הוצעו אף הם לתיקון טשטושים בתמונות על ידי שימוש במידע חיצוני בדומה למאמר זה אך לא בהכרח על ידי רשת. ישנם שיטות רבות לתיקון טשטוש בתמונה, למדנו על כך בהרצאה. פילטר "ווינר" יכול לשמש לתיקון טשטוש עם צירוף ידע מוקדם או שערוך פונקציית הטשטוש במערכת. שימוש בלמידה של פונקציית הטשטוש ולאחר מכן להשתמש בפילטרים. נעשו מחקרים לפני המחקר הזה על תיקון טשטוש באמצעות רשת מסוג CNN [9]. פילטרים הופכים ידועים בשימוש עבור לוחות רישוי בהם הטשטוש הוא העיקר והרעש פחות גורם משפיע.

## השיטה המוצעת במאמר

המאמר מציע שיטה לתיקון טשטושים הנובעים ממספר גורמים שונים כמו תנועה של האובייקט, מיקוד לא נכון של העדשה, רעידה של המצלמה וכדומה. כותבי המאמר טוענים שעבור בעיות בהם מתעסקים בכמה קני מידה שונים, הפתרון והפרמטרים משתפים ביניהם פרמטרים בכל קנה מידה. זה יוצר יציבות ועוזר למנוע התאמה יתירה של הפתרון לתמונות מסוימות. הקונספט הזה כמעט ולא מיושם ברשתות נוירונים מלאכותיות מסוג CNN (Convolutional Neural Networks). כותבי המאמר מציעים לשתף פרמטרים גם ברשתות מסוג זה משום שזה חוסך כמות פרמטרים ולכן מקצר את זמן האימון ומאפשר שימוש במידע לאורך כל הרשת.

המטרה היא לשחזר את התמונה כך שתתקבל תמונה חדשה וחדה. הכיוון של המחקר הוא תיקון טשטוש בתמונה יחידה בצורה כזו שמשחזרים את התמונה החדה בעזרת מעבר בין רזולוציות שונות מהקטנות לגדולות. השיטה המוצעת היא שימוש ברשת נוירונים אשר מקבלת תמונות בגודל מסוים (במקרה הזה רצף של תמונות), מקטינה אותם ומחלקת אותם ל-3 קני מידה שונים ("סקלות") של גודל התמונה. שיטת הפירמידה היא כלי מוכר בשיטות ומשימות שחזור מטשטוש או יצירת תמונות [4,9]. כלומר יש כאן מעין תהליך של שינוי גודל התמונה תוך כדי ניבוי של תמונה חדשה מתמונה מטושטשת מכל קנה מידה כתת משימה. בין כל קנה מידה לקנה מידה ישנו חיבור של הפלט מהקנה מידה הקודם אל הקלט של הקנה מידה החדש (הפלט מהקנה מידה קודם עובר הגדלה כך שיתאים לרזולוציה של קנה המידה החדש בעזרת אינטרפולציה בי-לינארית).

החוקרים מציעים שימוש בשיטת הפירמידה בשילוב עם רשת מסוג RNN (Recurrent Neural Network), שיטות קודמות נעשו בעזרת CNN (Convolutional Neural Networks) ברובם. בנוסף ישנו שימוש בארכיטקטורת מקודד-מפענח (Encoder-Decoder) שידועה בשימושה בבעיות הפוכות [1,11,2]. המקודד יוצר מפות מאפיינים של המידע המוזן אליו והמפענח בונה מידע חדש מן מפות המאפיינים שהוזנו אליו מהמקודד. במקרה של המאמר הזה, המקודד בונה מפות מאפיינים מהתמונה המטושטשת והפלט של הקנה מידה הקודם והמפענח ינסה לבנות תמונה מתאימה שאינה מטושטשת בעזרת מפות המאפיינים. ה-RNN בא פה לידי ביטוי בכך שישנו בלוק LSTM (Long-Short-Term-Memory) הכולל תמיד תהיה LSTM מסוג קונוולוציה) מסוג קונוולוציה בין המקודד והמפענח בנוסף לשימוש בפלט מהקנה המידה הקודם כקלט נוסף עבור הקנה מידה הבא, הבלוק בעצם יוצר סוג של שימור ידע ושיתוף פרמטרים בין קני המידה השונים (בנוסף לכך שהפלט מהקנה המידה הקודם הופך לקלט עבור RNN). כלומר בין כל קנה מידה לקנה מידה מתקיים שיתוף פרמטרים, זה לא מתרחש בדרך כלל ב-CNN ויכול לעזור לאימון מהיר ויעיל יותר של הפרמטרים. היתרון שנוצר מכך הוא שניתן לחלוק את הפרמטרים של כל "הסקלות" ולא לחשב פרמטרים שונים עבור כל סקלה וכך לחסוך זיכרון, זמן אימון וזמן ניבוי. הסיבה לבחירת LSTM על פני GRU (Gated Recurrent Unit) היא על בסיס ניסוי וטעיה של כותבי המאמר בלבד, למרות שהניסוי לא פורסם במאמר.

המקודד-מפענח שהחוקרים משתמשים בו עושה שימוש ב-ResBlocks אשר עוזרים מאוד להימנע מדיפוזיה או התפוצצות של גרדיאנטים ומוביל להתכנסות מהירה וטובה יותר [6,8]. כך נוכל בעצם להעמיק את הרשת ולהשתמש בעוד שכבות. בנוסף ישנם "קשרי קפיצה" בין המקודד והמפענח, אלו קשרי קפיצה "טווח ארוך" אשר בעצם מחברים מפות מאפיינים של המקודד אל מפות מאפיינים של המפענח וכך בעצם להעביר מאפיינים משכבות מוקדמות למאוחרות. השיטה המוצעת במאמר היא ללא שימוש בשכבות מיצוע (Batch Normalization) בהתבסס על מאמרים קודמים [9] ובניגוד לשימוש המקורי בבלוקים [6]. המפענח הוא בעל שכבות דה-קונוולוציה שזוהי בעצם שכבת שחלוף של קונוולוציה שמטרתה להגדיל את הגודל המרחבי של מפות המאפיינים ולהוריד את מספר הערוצים. כל שכבות

קונוולוציה ברשת הם עם קרנל בגודל 5x5 עם 2 צעדים, בResBlocks יש צעד 1 עבור הקונוולוציה. הקונוולוציה בתא LSTM היא 3x3.

כמו שנאמר, הקלט של הרשת הוא רצף של תמונות מטושטשות והמטרה שלה היא לנבא את התמונה "הנקייה" של כל תמונה כרצף תמונות. הקלט נחתך בצורה רנדומלית ומוקטן לרבע מהגודל המקורי. כלומר ניתן לרשום זאת כמשוואה:

$$I^i, h^i = \text{Net}_{SR}(B^i, I^{i+1\uparrow}, h^{i+1\uparrow}; \theta_{SR})$$

כאשר  $I$  זו התמונה המשוחזרת משוערכת,  $h$  אלו וקטורים פנימיים ברשת מתוך בלוק LSTM,  $B$  זו התמונה המטושטשת, "תטא" אלו פרמטרי האימון ברשת. הווקטורים הפנימיים הם מאפיינים מסקלות קודמות. פונקציית האקטיבציה של הרשת היא ReLU. ה"פותר" של הרשת הוא Adam עם פונקציית הפסד  $L_2$  כאשר קצב הלימוד דועך אקספוננציאלית.

$$L_2 = \sum_{i=1}^n \frac{k_i}{N_i} \|I^i - I_{ground\ truth}^i\|_2^2$$

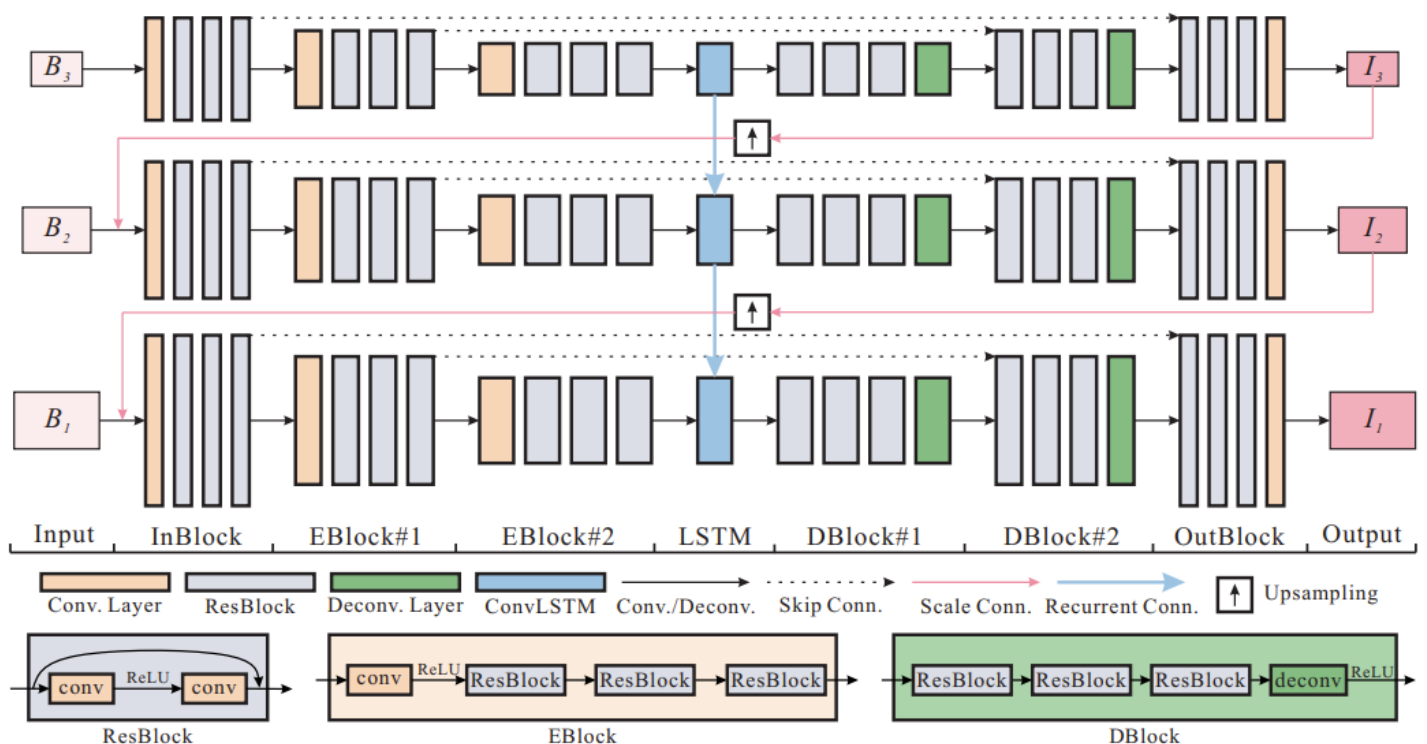
הקישור בין הקני מידה השונים הוא גם מתבצע מכך שמעבירים את הקלט של סקלה מסוימת לאחרת וגם על ידי קישור ב"צוואר הבקבוק" של הרשת כפי שנאמר בLSTM. ניתן להביע את המקודד-מפענח בצורה המתמטית הבאה:

$$f^i = \text{Net}_E(B^i, I^{i+1\uparrow}; \theta_E)$$

$$h^i, g^i = \text{ConvLSTM}(h^{i+1\uparrow}, f^i; \theta_{LSTM})$$

$$I^i = \text{Net}_D(g^i; \theta_D)$$

כאשר השורה הראשונה היא המקודד, השנייה שכבת LSTM והשלישית היא המפענח. ניתן לראות את העברת הפרמטרים בין השכבות, למשל הLSTM מקבל את תוצאת המקודד, המפענח מקבל את תוצאת הLSTM והLSTM בסקלה הבאה יקבל את וקטור הערכים החבויים של הLSTM מהסקלה הנוכחית. זוהי התמונה המשוערכת החדשה  $B_i$  זוהי התמונה המטושטשת,  $i+1$  מסמל את הקנה מידה הקודם שהוא חצי בגודלו מהנוכחי. החץ למעלה מסמן אינטרפולציה בי-לינארית.  $h$  אלו ערכים חבויים של הLSTM שמועברים אל התא הבא ומכילים מידע כמו פרמטרים והיסטוריית החישוב.



איור 1: תיאור הרשת בצורה סכמטית,  $B$  זו התמונה המטושטשת לפי סקלות (1-3),  $I$  זו התמונה המשוערכת החדשה. ניתן לראות את "קפיצות ארוכות טווח" בין המקודד למפענח בשכבות השונות בהתאמה ואת הקישור בין הLSTM.

המידע אשר נעשה בו שימוש לאמן את הרשת מורכב מתמונות של מצלמה עם חשיפה מהירה אשר נעשה מיצוע על רצף של תמונות כדי לקבל תמונה מטושטשת. בעצם מתקבלת תמונה שמדמה חשיפה ארוכה ולכן מטושטשת, זה מדמה מצבים אמיתיים יותר טוב מאשר הכפלה במטריצת טשטוש. גודל מאגר המידע הוא 3214 תמונות כזוגות של תמונה מטושטשת ותמונה נקייה. גודל החבילה בכל איטרציה ברשת הוא 16 תמונות אשר נחתכות בצורה רנדומלית לגודל של 256 על 256 פיקסלים. כל הפרמטרים ברשת אופסו בשיטת "אקסבייר" פרט לLSTM שהוא מתחיל מאפסים.

## תוצאות

את מדידת איכות התוצאה מדדו בעזרת PSNR וSSIM:

$$PSNR(f, g) = 10 \log_{10} (255^2 / MSE(f, g)) \quad SSIM(f, g) = l(f, g) c(f, g) s(f, g)$$

$$l(f, g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \quad s(f, g) = \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \quad c(f, g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \quad MSE(f, g) = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (f_{ij} - g_{ij})^2$$

כאשר ככל שהערך של שניהם עולה ככה איכות התמונה טובה יותר. החוקרים בדקו כמה מודלים שונים: רשת אשר משתמשת בקנה מידה אחד בלבד (SS) ללא LSTM. רשת עם קנה מידה משתנה אך לא מחובר בצורה שמשתפת פרמטרים, כלומר כל קנה מידה הוא רשת בפני עצמו (SC). רשת ללא קשרי RNN באמצע הרשת (w/oR). רשת אשר משתמשת הRNN קלאסי במקום LSTM (RNN). רשת אשר לא עושה שימוש במקודד-מפענח (SR-Flat), יש במקום זה 43 שכבות קונוולוציה רציפות. רשתות אשר משתמשות במספר שונה של ResBlocks הקרויות SR-EDRB עם מספר עוקב כמספר הבלוקים. רשת אשר בבלוקים שלה אין צעדים בכלל (SR-RB). רשת מקודד-מפענח ללא ResBlocks (SR-RD). ניתן לראות תוצאות של הרשת באיור 4 ואיור 5.

טבלה 1: מתארת את תוצאות הניסוי על רשתות שונות. השיטה המוצעת היא SR-EDRB3. ניתן לראות שלה יש את הערכים הטובים ביותר מבחינת איכות תמונה משוחזרת.

Model	SS	SC	w/o R	RNN	SR-Flat
Param	2.73M	8.19M	2.73M	3.03M	2.66M
PSNR	28.40	29.05	29.26	29.35	27.53
SSIM	0.9045	0.9166	0.9197	0.9210	0.8886
Model	SR-RB	SR-ED	SR-EDRB1	SR-EDRB2	SR-EDRB3
Param	2.66M	3.76M	2.21M	2.99M	3.76M
PSNR	28.11	29.06	28.60	29.32	<b>29.98</b>
SSIM	0.8991	0.9170	0.9082	0.9204	<b>0.9254</b>

בהשוואה לעבודות אחרות (עבודות קודמות לנוכחית) הרשת מראה שיפור גם באיכות התוצר וגם בזמן ההרצה:

טבלה 2: השוואה מול עבודות קודמות, ניתן לראות שישנו שיפור משמעותי מבחינת זמן הרצה ויחס אות-רעש.

Method	GOPRO		Köhler Dataset		Time
	PSNR	SSIM	PSNR	MSSIM	
Kim <i>et al.</i>	23.64	0.8239	24.68	0.7937	1 hr
Sun <i>et al.</i>	24.64	0.8429	25.22	0.7735	20 min
Nah <i>et al.</i>	29.08	0.9135	26.48	0.8079	3.09 s
Ours	<b>30.26</b>	<b>0.9342</b>	<b>26.75</b>	<b>0.8370</b>	<b>1.87s</b>

התוצאות מהמאמר נראות יפות, המאמר מוסבר ברור ולעניין ומצורף אליו קוד פתוח לעיון ושחזור. מאז הוצאת המאמר התפרסמו כמה וכמה מחקרים בנושא והם מסתכלים על המאמר הזה State of the art ובסיס לבחינת השיטה שלהם [7,16]. בנוסף ישנו מאמר חדש שמשפר את השיטה שלהם על ידי טשטוש גדול יותר של התמונה [10].

ייתכן וישנה בעיה שנוצרת מכך שיש שיתוף פרמטרים כולל בין קני המידה השונים, יש מאפיינים ספציפיים לכל סקלה וייתכן ונאבד אותם. למשל מאפיינים באזורים חדים הם דומים ואילו מאפיינים באזורים מטושטשים הם שונים בין קנה מידה אחד לאחר (ניתן לראות את זה כאשר מקטינים ומגדילים תמונה). לעיתים כשמקטינים תמונה היא נראית מעט חדה יותר וייתכן שכאשר הרשת לומדת מאפיינים בקנה המידה הנמוך היא לומדת יותר מאפיינים של חדות מאשר טשטוש. עבודה מעט חדשה יותר בנושא של כותבי המאמר מראה שישנו יתרון לשיתוף פרמטרים ברשת דומה ללא LSTM כאשר השיתוף מתבצע בצורה קצת שונה [5]. הכותבים מראים שיפור של 1 דציבל בPSNR כאשר בין השכבות המוציאות מאפיינים במקודדים אין שיתוף פרמטרים אך בין השכבות של הטרנספורמציות לא-לינאריות כן יש שיתוף פרמטרים. הכוונה בשכבות מוציאות המאפיינים היא לשכבות קונוולוציה לפני ResBlocks ואילו הבלוקים הם הטרנספורמציות הלא-לינאריות. הניסוי הזה מאריך את זמן ההוצאה לפועל של הרשת ב-0.3 שניות כתוצאה מהגדלת הפרמטרים אך מעלה את איכות התוצאה מבחינת הורדת הטשטוש בתמונה.

## סיכום

כותבי המאמר הציגו שיטה ששוחזרה על ידי חוקרים אחרים ואף שופרה [5,10], השיטה נבחנה גם על תמונות מהעולם האמיתי (איור 5), הכוונה לתמונות אשר לא טושטשו בצורה מלאכותית וניתן לראות תוצאות יפות. כותבי המאמר הסבירו את השיטה היטב ושיתפו את הקוד שלהם כך שחוקרים אחרים יוכלו לשפר ולשחזר. נעשו מספר ניסויים על מנת להראות שסידור הרשת בצורה המוצגת הוא אכן הגיוני. קצת מוזר לי שבניסוי שלהם LSTM תרם הרבה ואילו בניסוי של אותה קבוצת מחקר על שיפור השיטה המוצגת, לא נעשה בו שימוש ואף כלל לא הוזכר [5].

הצורה של הרשת מאוד מעניינת, נראה ששילוב שיטת הפירמידה עם קני מידה שהולכים וגדלים, יחד עם קישוריות שמוסיפה את הקלט לפלט של שכבה מסוימת והעברה של פרמטרים לקנה מידה הבא, יצרה תהליך מאוד מעניין של שחזור התמונה. גם עצם ההתמקדות שלהם בצורת הרשת עצמה ללא התעסקות גדולה בפונקציית ההפסד מעניינת מאוד, ממה שאני רואה בדרך כלל עושים השוואות ומציגים כל מיני "פותרים" וקצבי למידה שונים. כאן הם בחרו מעין "ברירת מחדל" מוצלחת עם פרמטרים מומלצים על פי המאמר שהציג אותה לראשונה.

לצערי טרם יכולתי לשחזר את הניסוי שלהם (כן ניסיתי, אך סביבת העבודה שהחוקרים השתמשו בה דרשה יותר מדי התאמות והייתי צריך לזנוח את זה מחוסר זמן), בעתיד אנסה לשחזר את המאמר בשביל עבודת המחקר שלי יחד עם מאמרי ההמשך שהקבוצה הוציאה ששיפרו את השיטה. יכול לשמש אותי בהורדת טשטוש יחסית כללית ומהירה כאשר האובייקט המצולם זז מהר (תרנגולות למשל יחסית זזות מהר כי זו חיה קטנה). קבוצת המחקר הזו מוציאה מאמרים מעניינים וכדאי לשים אליה לב.

כמה הסברים על נושאים שהרגשתי צורך להתעמק בהם כדי להבין את המאמר. צמצמתי את ההסבר עליהם כמה שאפשר.

### Convolutional Neural Networks

רשת CNN בעצם בונה מפות מאפיינים על ידי למידה של הקלט המוזן אליה בצורה של מכפלה סקלרית של חלון כלשהו על הקלט שמוזן אליה, במקרה שלנו הוא רצף של תמונות. שכבת קונבולוציה בעצם תחשב מכפלת נורונים המחוברים לאזורים מקומיים בתמונה ובין המשקולות ובין האזור שאליו הם מחוברים בתמונה. אם נניח נבחר עומק של 12 פילטרים נקבל מבנה תלת-ממדי של 12 מפות מאפיינים על גודל התמונה שנניח שהוא 256 על 256 ונקבל  $12 \times 256 \times 256$ . כל פילטר בעצם עובר על התמונה ומבצע קונבולוציה על אזור בה כגודל הקרנל (החלון אשר נבחר, נניח  $5 \times 5$ ) ובונה מפת דו-ממד שהיא הפילטר הנלמד. כלומר רשתות CNN מפעילות פעולות על התמונה המקורית ויוצרות שכבות של מפות מאפיינים וככל שהרשת עמוקה יותר נלמד יותר מאפיינים. כאשר מכפילים בפונקציית האקטיבציה נקבל תוצאה לא לינארית שתתן לנו את המאפיין הנלמד מודגש טוב יותר.

### Encoder-Decoder

מדובר במבנה שבדרך כלל סימטרי והוא ממפה מידע הנכנס בצורה רציפה לוווקטור במקודד והמפענח בתורו לוקח את הוווקטור ובונה תוצאה חדשה של רצף מידע (במקרה שלנו תמונות). המקודד יכול לקבל מידע באורך מסוים והמפענח יכול להוציא מידע באורך שונה לחלוטין.

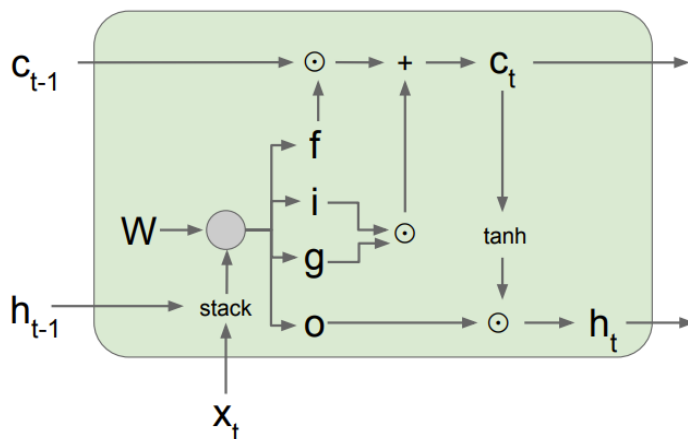
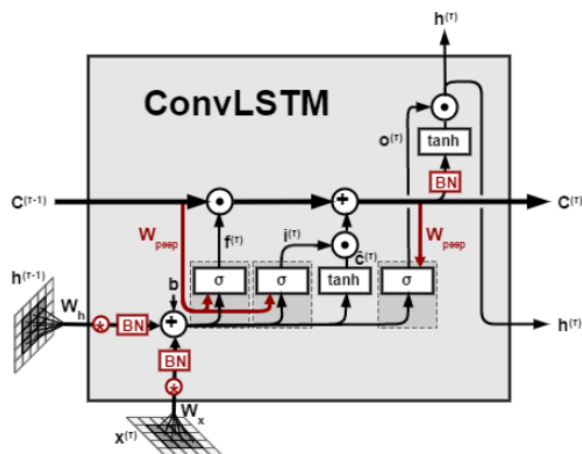
### Recurrent Neural Network

ב-CNN הניבוי של עצם אחד לא בהכרח מתקשר לניבוי של עצם אחר, כלומר אם המידע נכנס בצורה רציפה אין זיכרון במערכת ללמוד ממה שהיה בעבר. RNN מבצע את אותה המשימה עבור כל אלמנט בסדרה שנכנסה כאשר הפלט תלוי בחישובים קודמים. ל-RNN יש משקולות משותפות לאורך התהליך ולכן יכול "לזכור" מידע שנצבר. RNN נוטה לגרום לגרדיאנטים להתפוצץ או להיעלם יותר מהר מ-CNN לכן נעשה שימוש בעוד תוספות בדרך כלל וקשרי קפיצה "קצרי או ארוכי" טווח. הסיבה לכך היא שב-RNN הפרמטרים משותפים לעומת CNN, מה שגורם לרשת ללמוד יותר מהקלט שהיא ראתה לאחרונה בהשוואה לקלט הראשון שהיא ראתה.

### Long-Short-Term-Memory

נועד לפתור את הבעיה שנוצרת ב-RNN כאשר גרדיאנטים גדלים מאוד או קטנים מאוד ולכן נעלמים או מתפוצצים וחיזוי משפט למשל לומד יותר מסוף המשפט מאשר תחילת משפט. במאמר משתמשים בתא מסוג קונבולוציה אשר ההבדל בינו לבין הרגיל הוא שנעשית פעולה של קונבולוציה במקום כפל. בנוסף בתא מסוג זה המידע מתחלק על פני קוביית תל ממד. שיטה נוספת שמשתמשים בה כתוספת היא קיצוץ גרדיאנטים. יש 3 שערים, שער שכחה, שער קלט ושער פלט. שער השכחה מחליט כמה מה"עבר" צריך לזכור על ידי פונקציית סיגמואיד ועל פי המצב של התא הקודם והקלט הנוכחי מוציא ערך החלטה בין 0 ל-1 עבור תא הקודם. שער הקלט מחליט כמה מהיחידה הנוכחית יתווסף למצב הנוכחי בתא. על ידי סיגמואיד וגם טנגנס היפרבולי נקבל משקול בין 1 ל-1 שקובע כמה יש חשיבות לקלט עבור התא הנוכחי להעביר הלאה. שער פלט מחליט איזה חלק במצב התא הנוכחי יועבר לפלט על פי חשיבות. C זה תא הזיכרון הנצבר, הוא צובר את מצב האינפורמציה הנוכחית. התא נכתב, מוגש ונערך על ידי פרמטרים מוגדרים ושערים [12].

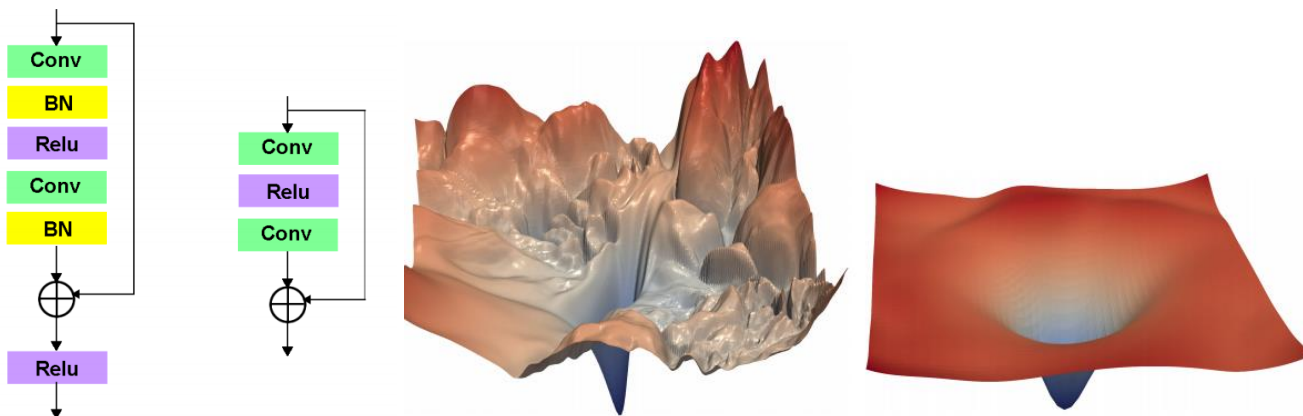
כעת, עבור תא מסוג קונבולוציה, ניתן להתייחס גם לפרמטרים המרחביים בכל שכל קלט, מצב, שער ומצב חבוי הם בצורת מידע ספקטרלי תלת ממדי כך שניתן להתייחס למידע דו ממדי. הטנגנס היפרבולי דואג לפיזור גרדיאנטים כדי למנוע פיצוץ או איפוס.  $h$  זוהי מטריצה של ערכים חבויים שהם בעצם הפלט של תאים קודמים.  $b$  זה  $BIAS$ ,  $Wx$  זה מטריצת משקולות.



איור 2: בצד ימין ניתן לראות בלוק LSTM רגיל ואילו בצד שמאל נראה בלוק LSTM מסוג קונוולוציה.

## ResBlocks

אלו בעצם בלוקים שבנויים כך שיש מספר שכבות קונוולוציה, שביניהם יש פונקציית אקטיבציה (ReLU למשל) ומיצוע כלשהו והעיקר כאן הוא החיבור בקשר "טווח קצר/מדלג" בין הקלט שנכנס לבלוק והפלט של הבלוק. כלומר הקלט ללא השינוי מתווסף לתוצאה של הבלוק. היתרון שמתקבל הוא יציבות והתכנסות מהירה יותר של הפתרון שמחפשים.



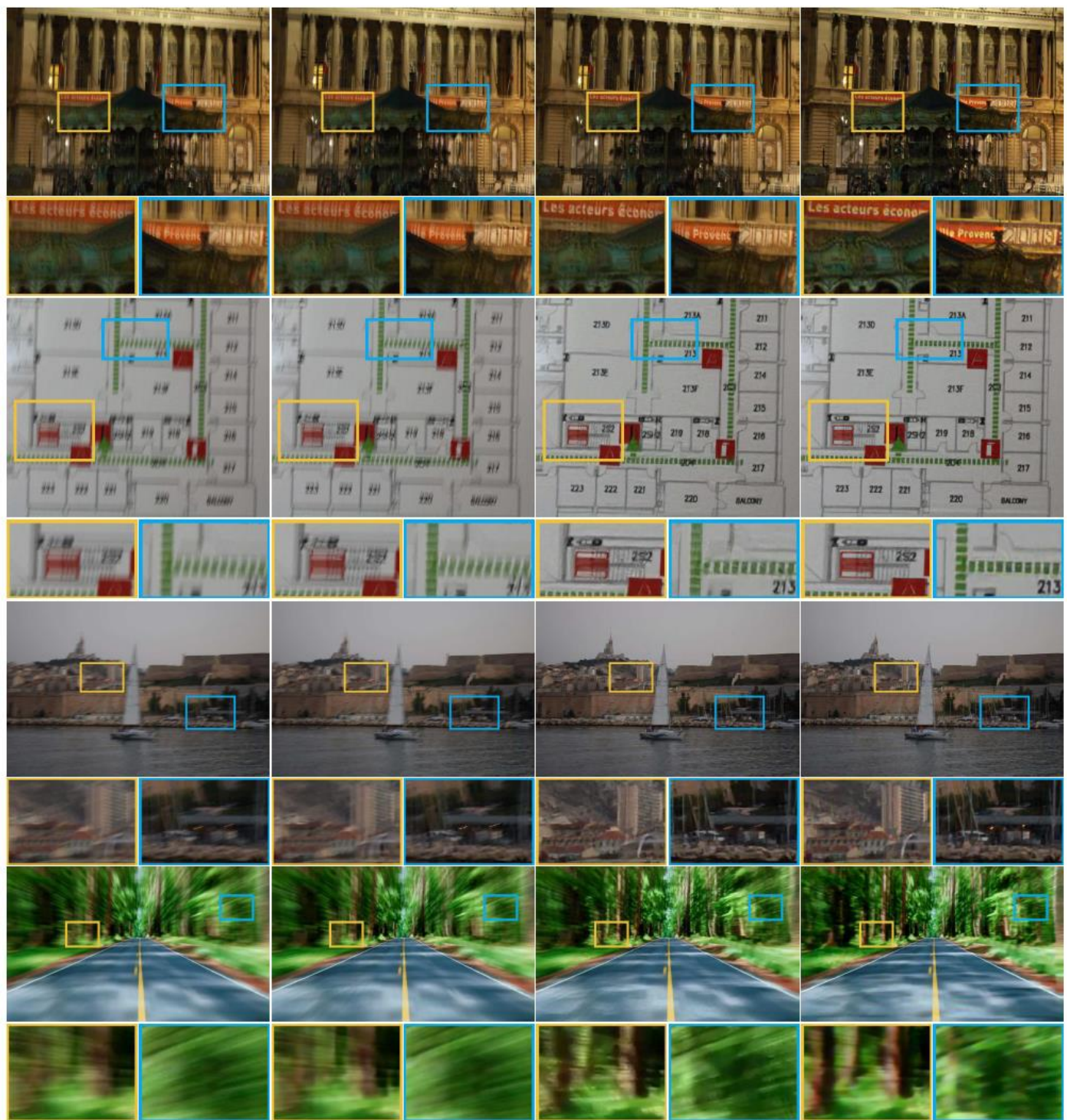
איור 3: משטח פונקציית ההפסד בשני האזורים הימניים, צד ימין עם קשרי קפיצה וצד שמאל ללא קשרי קפיצה. 2 האזורים השמאליים הם דוגמה לבלוק, כאשר הימני מבניהם (ללא שכבות המיצוע/נרמול) הוא האחד שחוקרי המאמר עשו בו שימוש.





איור 4: תמונות של סט המבחן. עמודה הכי שמאלית (a) היא של [15], עמודה (b) היא של [13], עמודה (c) היא של [9] והעמודה (d) היא של המאמר הנוכחי.





(a) Input

(b) Sun et al.

(c) Nah et al.

(d) Ours

איור 5: תוצאות המאמר מול עבודות אחרות עם תמונות "מהעולם האמיתי", כלומר תמונות מטושטשות לא מלאכותיות. עמודה הכי שמאלית (a) היא התמונה המקורית, עמודה (b) היא של [13], עמודה (c) היא של [9] והעמודה (d) היא של המאמר הנוכחי.

1. Bahat, Yuval, Netalee Efrat, and Michal Irani. "Non-uniform blind deblurring by reblurring." Proceedings of the IEEE International Conference on Computer Vision. 2017.
2. Cho, Sunghyun, and Seungyong Lee. "Fast motion deblurring." ACM SIGGRAPH Asia 2009 papers. 2009. 1-8.
3. Cho, Sunghyun, Yasuyuki Matsushita, and Seungyong Lee. "Removing non-uniform motion blur from images." 2007 IEEE 11th International Conference on Computer Vision. IEEE, 2007.
4. Denton, Emily, et al. "Deep generative image models using a laplacian pyramid of adversarial networks." arXiv preprint arXiv: 1506.05751 (2015).
5. Gao, Hongyun, et al. "Dynamic Scene Deblurring with Parameter Selective Sharing and Nested Skip Connections." Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2019-June, 2019, pp. 3843–51, doi:10.1109/CVPR.2019.00397.
6. He, Kaiming, et al. "Deep Residual Learning for Image Recognition." Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2016-Decem, 2016, pp. 770–78, doi:10.1109/CVPR.2016.90.
7. Kupyn, O., Martyniuk, T., Wu, J. & Wang, Z. DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. in Proceedings of the IEEE International Conference on Computer Vision 2019-October, 8877–8886 (Institute of Electrical and Electronics Engineers Inc., 2019).
8. Li, H., Xu, Z., Taylor, G., Studer, C. & Goldstein, T. Visualizing the loss landscape of neural nets. Adv. Neural Inf. Process. Syst. 2018-December, 6389–6399 (2018).
9. Nah, Seungjun, Tae Hyun Kim, and Kyoung Mu Lee. "Deep multi-scale convolutional neural network for dynamic scene deblurring." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
10. Park, D., Kang, D. U. & Chun, S. Y. Blur More To Deblur Better: Multi-Blur2Deblur For Efficient Video Deblurring. (2020).
11. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 9351, 234–241 (2015).
12. Shi, X. et al. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. in Advances in Neural Information Processing Systems vols 2015-Janua 802–810 (2015).
13. Sun, Jian, et al. "Learning a convolutional neural network for non-uniform motion blur removal." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.
14. Tao, Xin, et al. "Scale-recurrent network for deep image deblurring." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018.
15. Whyte, Oliver, et al. "Non-uniform deblurring for shaken images." International journal of computer vision 98.2 (2012): 168-186.
16. Zhang, K. et al. Deblurring by Realistic Blurring. in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2734–2743 (2020). doi:10.1109/CVPR42600.2020.00281.