
هو الحق



گزارش کار شماره ۱ پروژه کارشناسی

دانشکده ی فنی مهندسی

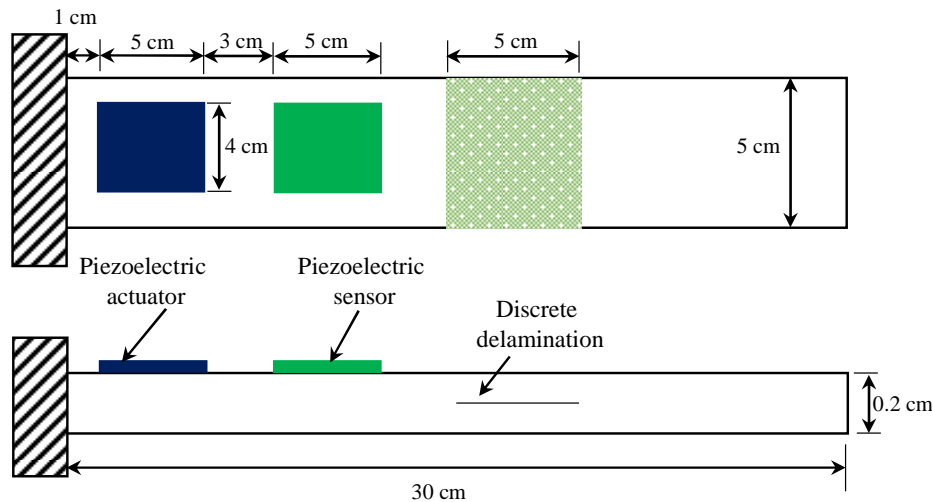
نام نویسنده:

عرشیا نیک بخت

مهر ۱۴۰۰

فهرست مطالب

۳	Structural Vibration-Based Classification and ...	i
۳ ساختار آزمایش	۱..i
۴ پیش پردازش داده ها	۲..i
۶ پیاده سازی یادگیری نظارت شده برای طبقه بندی داده های ورودی به کمک الگوریتم های	۳..i
۶ یادگیری ماشین	۴..i
۶ پیدا کردن تورم به کمک یادگیری عمیق	
۱۰	Full wavefield processing by using FCN ...	ii
۱۱ پیش پردازش	۱..ii
۱۲ پیدا کردن محل و اندازه تورم با روش یادگیری عمیق	۲..ii



شکل ۱: ساختار فیزیکی آزمایش برای پیدا کردن تورم روی قطعه های کامپوزیتی

1. Structural Vibration-Based Classification and ...

در این مقاله به کمک طبقه بندی^۱ می‌خواهیم محل تورم^۲ را تشخیص بدهیم.

توجه: در این روش برای پیدا کردن اندازه تورم استفاده نمی‌شود.

مقاله ای که می‌خواهیم مورد بررسی قرار بدیم از ارتعاش فرکانس پایین در خروجی سنسور پیزوالکتریک^۳ استفاده می‌شود و به همین دلیل روش ساده و بهینه ای برای پیدا کردن محل تورم می‌باشد. از محدودیت ها و مضایای این روش در ادامه صحبت خواهیم کرد. در ابتدا باید با ساختار روش انجام آزمایش آشنا بشیم و سپس به روش های اعمال یادگیری عمیق^۴ و CNN^۵ به روی ساختار آزمایش گفته شده بحث خواهیم کرد.

۱.۱. ساختار آزمایش

همانطور که در شکل ۱ می‌بینید آزمایش ما از دو پیزوالکتریک سنسوری و عملگری در دو محل متفاوت ساخته شده است و همینطور تورم ما که محل آن می‌تواند متغییر باشد. محل اعمال سنسور و عملگر را می‌توان به کمک آزمون خطا بدست آورد. به صورتی که هدف ما ایجاد بیشترین تغییرات فرکانس ارتعاشی هنگام اعمال ارتعاشات می‌باشد (که به آن reward می‌گویند) و راهکار های ما برای اینکار قرار دادن عملگر و سنسور در قسمت های مختلف سطح قطعه کامپوزیتی می‌باشد (که به آن actions می‌گویند). هدف ما گرفتن بیشترین reward از action های انجام شده می‌باشد برای اینکار با ایجاد جدول یا ماتریس هایی از actions و با کانال هایی از rewards انجام میشود که میتوان بهترین action را به کمک ساده ترین الگوریتم یادگیری تقویتی^۶ انجام بدهیم که به آن Multi-Armed Bandits می‌گویند. راجب آن بدلیل اینکه صورت مسئله اصلی ما نیست بیشتر توضیح نخواهیم داد.

طبقه بندی ما به چه صورت خواهد بود؟

طبقه بندی ما به صورت اعمال تورم های متنوع به روی قطعه های مختلف خواهد بود به صورتی که در شکل ۲ می‌توانید ببینید هر قطعه را با یک نوع تورم برای داده برداری در اختیار خواهیم داشت که در چهار عمق متفاوت (D_0, D_2, D_4, D_6) روی قطعه کامپوزیتی روی ۳ محل

^۱ Classification

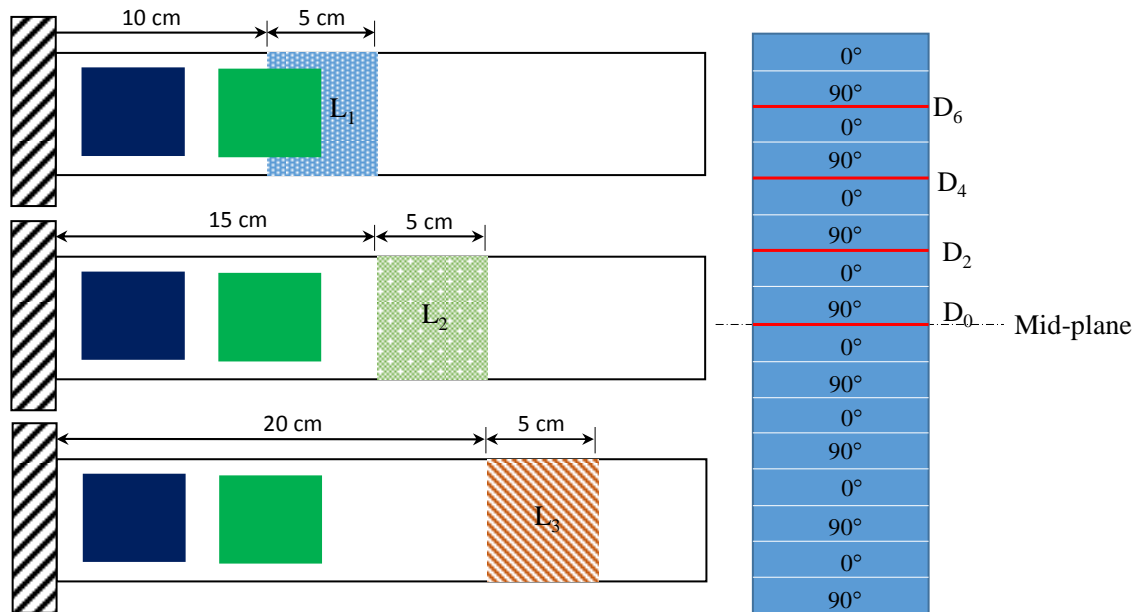
^۲ delamination

^۳ Piezoelectric sensor برای دریافت ارتعاش های وارد شده استفاده می‌شود. به این صورت کار میکند که با اعمال ارتعاش بر روی آن مقاومت الکتریکی آن تغییر می‌کند به صورتی که با بدست آوردن ولتاژ دو سر آن می‌توان مقدار ارتعاشات اعمال شده را بدست آورد که از همه مهمتر فرکانس ارتعاشات وارد شده.

^۴ Deep Learning

^۵ convolutional Neural Network

^۶ Reinforcement Learning



شکل ۲: روش طبقه بندی های قطعه های کامپوزیتی برای داده برداری فرکانس های طبیعی آن ها.

متفاوت حول (L_1, L_2, L_3) قطعه اعمال شده است، بعلاوه قطعه ای که هیچ توری روی آن قرار ندارد یعنی ۱۳ امین قطعه کامپوزیتی در اختیار داریم که ۱۲ تای آن ها از $L_1 D_0$ تا $L_3 D_6$ طبقه بندی شده اند، بعلاوه قطعه سالم که آن را H نامگذاری کردیم. دیتای ورودی ما برای پردازش به صورت 13000×1001 می باشد که 13000 برای ۱۰۰۰ نمونه تحریک هارمونیک رندم برای ۱۳ قطعه مد نظر می باشد و ۱۰۰۱ بیانگر ۱۰۰۱ پله زمانی از 0 تا 0.1 ثانیه تقسیم شده است. پس می توان به این صورت خلاصه نمود که ما دارای ۱۳۰۰۰ نمونه با ۱۰۰۱ ویژگی از هر نمونه را در اختیار داریم.

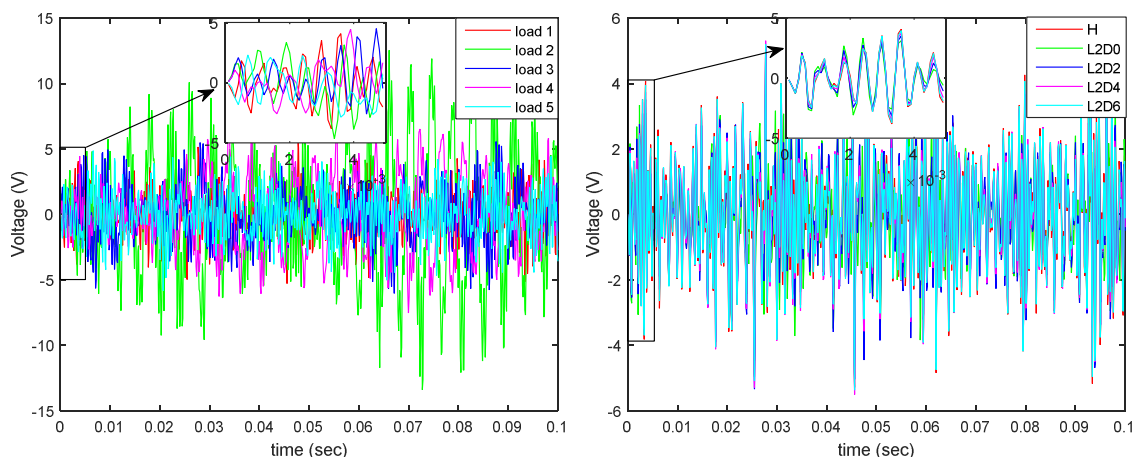
۲.۱. پیش پردازش داده ها

ابتدا قبل از وارد شدن به یادگیری ماشین، باید یک سری پیش پردازش هایی را روی داده های ورودی انجام بدیم برای طبقه بندی بهتر داده ها توسط ماشین برای اینکار ابتدا باید شکل کشیده شده از داده های ورودی برای نمونه های مختلف را ترسیم بکنیم که می توانید آن را در شکل ۳ ببینید.

با توجه به شکل ۳ می توان دریافت که متمایز کردن تغییرات ولتاژ بر روی قطعه هایی با تورم های متفاوت از تحریک هارمونیک ما کار بسیار سختی می باشد به همین منظور ما نیاز به پیش پردازش برای روشن کردن این تمایز هستیم. برای اینکار ما با کمک الگوریتم FFT فرکانس ارتعاشات خروجی را برحسب اندازه آن ها رسم میکنیم. از الگوریتم FFT^۷ معمولاً برای نویز گیری صدا ها و تصویر ها انجام می شود. روش این الگوریتم درواقع بهینه شده DFT^۸ می باشد. ابتدا برای فهم کلی این روش الگوریتم DFT را توضیح میدهم و سپس به FFT می پردازم.

ابتدا دلیل نیاز بشر به تبدیل فوریته ماتریس های ورودی را مورد بررسی قرار می دهم. برای مثال صدای ضبط شده یک رکورد در قطعه دارای نویز زیادی از محیط اطراف می باشد و وظیفه رکورد پیدا کردن فرکانس صدای کاربر درحال ضبط می باشد و برای اینکار ابتدا با دریافت فرکانس صدای کاربر و تمایز آن با فرکانس های محیط عمل فیلتر کردن را انجام بدهد. پس نیاز داریم که اندازه فرکانس های دریافتی توسط میکروفون را پیدا کیده و سپس فرکانس های برجسته (فرکانس های برجسته دارای اندازه فرکانس بیشتری می باشند) را با دیگر فرکانس ها با اعمال threshold به آن ها خارج کرده و سپس با تبدیل فوریته معکوس آن را پخش کنیم. تبدیل فوریته را از ریاضی مهندسی خواندیم که با جمع اعداد مختلط به تقریب خوبی از خود تابع می رسیدیم و می دانیم که اعداد مختلط دارای اندازه و فاز هستن پس با تبدیل فوریته اعداد ورودی به فرکانس و اندازه آن ها می رسمیم (برای مشاهده مثال گفته شده می توانید به یکی از پروژه های بنده در گیت هاب مراجعه فرمایید).

^۷Fast Fourier Transform
^۸Discrete Fourier Transform



شکل ۳: دیتا های ورودی گرفته شده به صورت خام از پیزوالکتریک که به صورت ولتاژ برحسب زمان می باشد. شکل سمت راست نمودرا نمایش داده شده ۱ تحریک هارمونیک بر روی ۱۳ قطعه می باشد و همینطور نمودار سمت چپ برای تحریک های های رندم هارمونیک بر روی قطعه سالم را مورد نمایش قرار می دهد.

تبدیل فوریه ای که از قبل خواندیم به صورت زیر نمایش داده می شود:

$$\hat{f}_k = \sum_{j=0}^{n-1} f_j e^{-i2\pi j \frac{k}{n}} \quad (1)$$

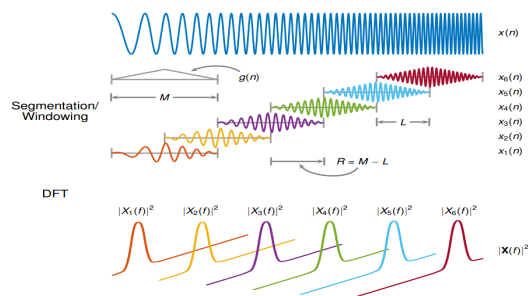
حال اگر ما عدد ثابتی به نام W_n تعریف کنیم:

$$W_n = e^{-2\pi \frac{i}{n}} \quad (2)$$

حال تبدیل فوریه f_k به \hat{f}_k به صورت زیر نمایش داده می شود.

$$\begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{n-1} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & W_n & W_n^2 & \cdots & W_n^{n-1} \\ 1 & W_n^2 & W_n^4 & \cdots & W_n^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & W_n^{n-1} & W_n^{2(n-1)} & \cdots & W_n^{(n-1)^2} \end{bmatrix} \times \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_{n-1} \end{bmatrix} \quad (3)$$

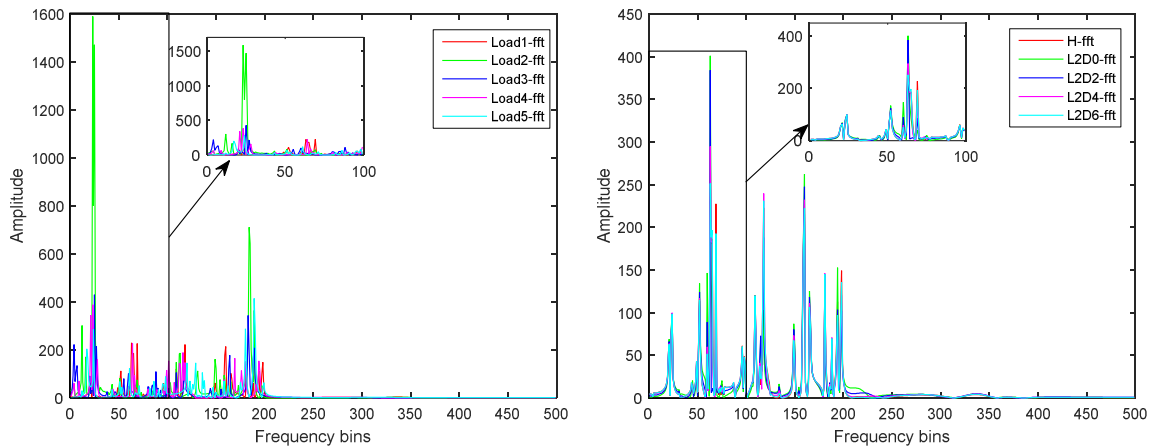
و حال ماتریس $\begin{bmatrix} \hat{f}_0 \\ \hat{f}_1 \\ \hat{f}_2 \\ \vdots \\ \hat{f}_{n-1} \end{bmatrix}$ همگی اعداد مختلط بوده که دارای فاز و اندازه می



شکل ۴: برای فهم روش DFT به شکل بالا توجه فرمایید.

باشند و با جمع مقادیر رسم شده آن ها به شکل تابع اولیه می رسمیم. حال ما با ترسیم اندازه برحسب فرکانس می توانیم پیش فرکانس های تاکید شده روی داده های ورودی را بدست آوریم.

همانطور که گفته شد الگوریتم FFT بهینه شده DFT می باشد و برای این می باشد که حجم محاسباتی الگوریتم DFT به اندازه n^2 در صورتی که n تعداد دیتا های ورودی می باشد، است. درحالی که الگوریتم FFT دارای حجم محاسباتی $n \log(n)$ می باشد. یعنی با زیاد شدن دیتا های ورودی حجم محاسباتی ما بدلیل لگاریتم افزایش خیلی زیادی نخواهد داشت (راجب الگوریتم آن بدلیل نبود هدف این گزارش کار توضیح نخواهم داد).



شکل ۵: نمودار سمت راست اندازه برحسب فرکانس اعداد مختلط برای ارتعاش ثابت به روی قطعه هایی با تورم های مختلف و سالم می باشد و نمودار سمت چپ اندازه برحسب فرکانس برای ارتعاشات متفاوت بر روی قطعه سالم کامپوزیتی است.

پس با انجام تبدیل فوریه می توان تمایز طبقه بندی های متفاوت را مشاهده نمود. به این صورت که هر تورم دارای اندازه بخصوصی در فرکانس مد نظر می باشد. هرچند که تمایز آنها بسیار ناچیز است ولی برای ورودی الگوریتم های ماشین ما مناسب می باشد.

۳.۱ پیاده سازی یادگیری نظارت شده برای طبقه بندی داده های ورودی به کمک الگوریتم های یادگیری ماشین

Classifier	Cubic SVM	Medium Gaussian SVM	J48 Decision Tree	Random Forest	Naïve Bayes	K-Nearest Neighbors	Recurrent Neural Network
Classification Accuracy	16.00 %	19.60 %	13.04 %	4.23 %	17.43 %	10.50 %	66.00 %

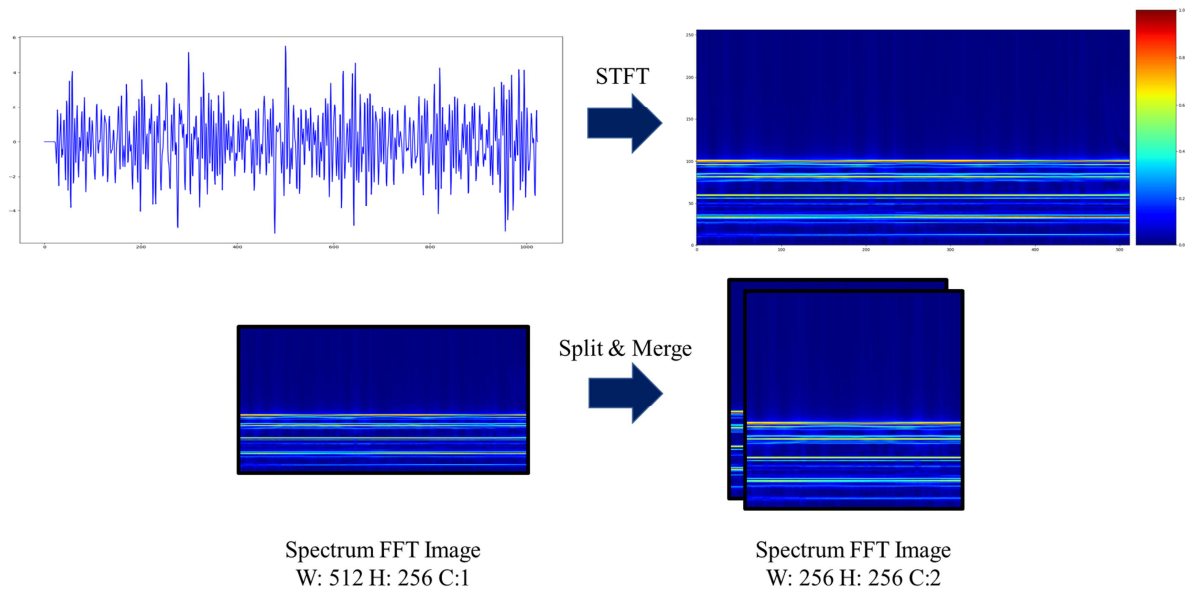
جدول ۱: دقت های خروجی از الگوریتم های ساده یادگیری ماشین.

همانطور که از جدول ۱ پیداست دقت خروجی مدل های یادگیری ماشین بسیار پایین است و متأسفانه فابل اعتماد برای ما نیستند پس ما نیاز به روش جدیدی برای اینکار داریم که در فرکانس های خروجی دیتا های ورودی ما عمیق تر شده و دقت بالاتری را به ما بدهد به همین دلیل وارد حوضه یادگیری عمیق می شویم.

۴.۱ پیدا کردن تورم به کمک یادگیری عمیق

برای اینکه بتوانیم به صورت عمیق از داده های ورودی استفاده کنیم باید از کانوها^۹ استفاده کنیم که برای اینکار دیتای ورودی باید به صورت ۲ بعدی باشد (البته کانوهای ۱ بعدی هم داریم ولی دارای دقت پایین تری می باشد) چرا که با تعداد پارامترهای محدود تری میتونیم ارتباط بین feature های ورودی را بهتر متوجه ماشین کنیم. به دلیل برتری CNN از NN^{۱۰} نمی پردازیم به اینکه NN دارای پایامترهای

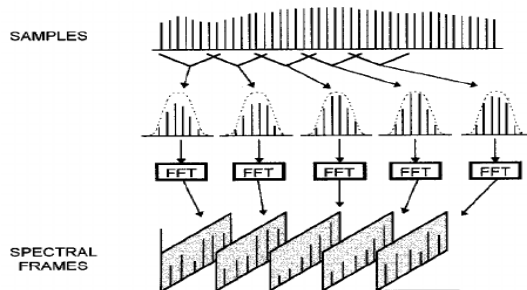
^۹Convolutional
^{۱۰}Neural Network



شکل ۷: پیش پردازش اعمال شده به روی دیتا های ورودی برای بدست آوردن طیف های رنگی ارتعاشات فرکانس پایین گرفته شده از سنسور پیزوالکتریک. دلیل نصف کردن عکس عای ۵۱۲ در ۲۵۶ به عکس عای ۲۵۶ در ۲۵۶ را می توان با راحتی ایجاد کانو های ۳ در ۳ را توجیح نمود.

قابل آموزش^{۱۱} بیشتری است ولی دید کلی آن بدلیل ارتباط دهی بهتر CNN بین ویژگی های نمونه های ورودی اشاره نمود. طبق سایت [Applied Deep Learning](#)^{۱۲} دلیل این مطلب را مشاهده کنید.

حال چگونه دیتای ورودی را از ۱ بعدی به دو بعدی تبدیل کنیم؟



شکل ۶: روش STFT

برای جواب این سوال نیاز داریم که با الگوریتم STFT^{۱۳} آشنا بشویم. روش این الگوریتم بسیار شبیه به روش قبلی می باشد با این تفاوت که ۰.۱ ثانیه را به تعداد پله های بسیار کوتاهی تقسیم کرده و سپس طیف رنگی^{۱۴} آن را مناسب با مقدار فرکانس گرفته شده از آبی به قرمز تغییر می دهیم. برای مثال برای تبدیل دیتای ورودی به عکس ۲۵۶ در ۵۱۲ می توانیم داده های زمانی را به ۵۱۲ قسمت مساوی تقسیم کنیم و سپس فرکانس خروجی را از ۰ تا بیشترین مقدار آن به ۲۵۶ قسمت تقسیم کنیم و طیف رنگی را بوجود بی آوریم.

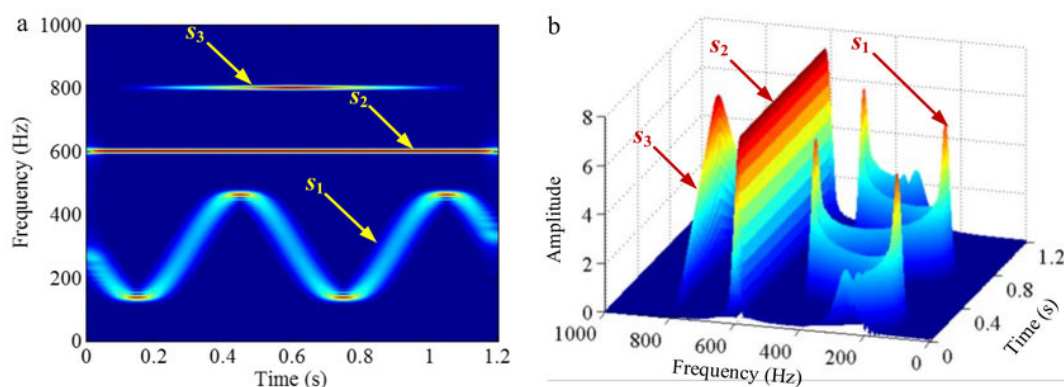
برای فهمیدن راحت تر این روش باید به شکل ۳ بعدی خروجی آن نگاه بندازیم که در شکل ۶ قابل مشاهده می باشد. می توان دریافت که چرا این روش دارای برتری بیشتری از FFT می باشد چرا که در شکل دو بعدی ایجاد شده ویژگی های زمان، فرکانس و اندازه به صورت یکجا قابل مشاهده بوده و به ماشین داده می شود. حال فهمیدیم که الگوریتم STFT چگونه کار می کند و الان نوبت اعمال آن به روی دیتا های ورودی می رسد که در شکل ۷ قابل مشاهده است.

و حال مدل شبکه عصبی مورد نیاز به صورت کاملاً واضح و کامل در مقاله آمده است به همین دلیل پیاده سازی مدل کار وقت گیری نخواهد بود و در شکل ۹ می توانید مدل پیاده سازی شده را مشاهده کنید.

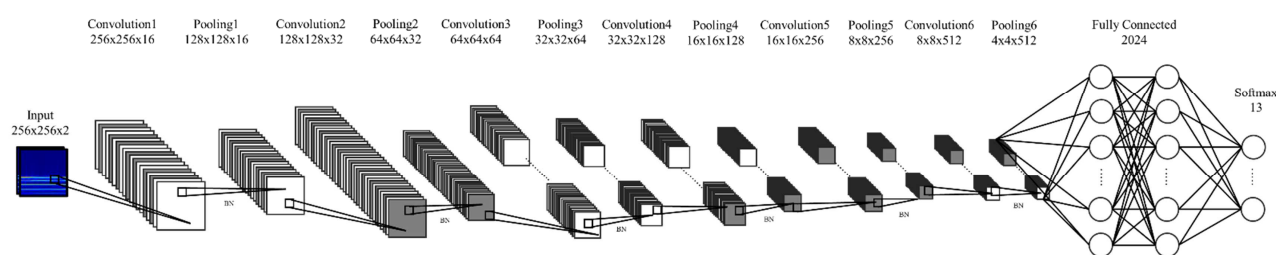
^{۱۱} trainable parameters را می توان همان w, b هایی در نظر گرفت که با آن می توان خروجی مدل را بدست آورد

^{۱۲} "The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision. For example, given many pictures of cats and dogs it learns distinctive features for each class by itself. CNN is also computationally efficient."

^{۱۳} Short-Time Fourier Transform
^{۱۴} spectrograms



شکل ۸: تصویر ۳ بعدی از روش STFT می باشد که نشان می دهد در زمان های متفاوت چه فرکانسی برجسته تر بوده است.



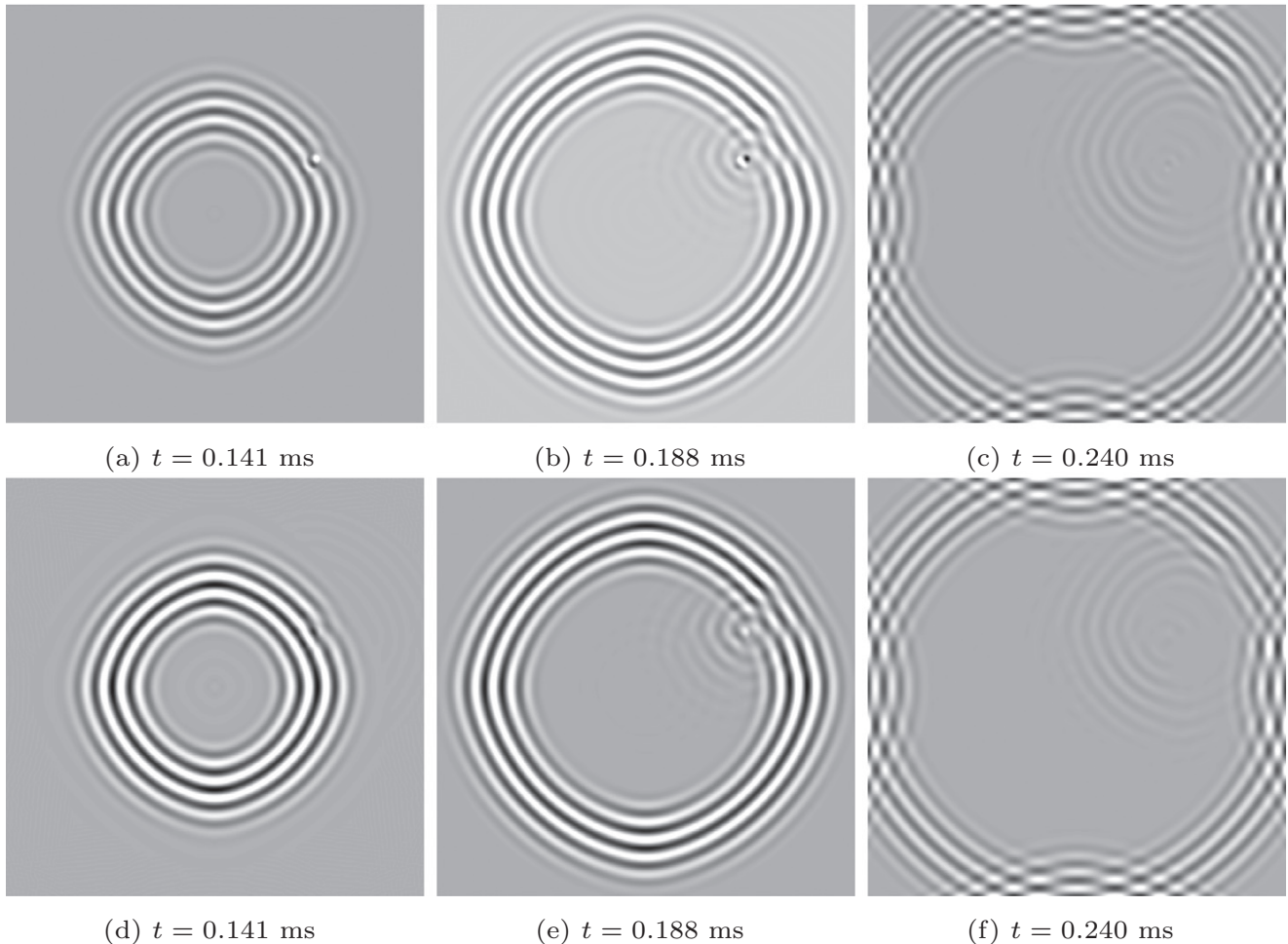
شکل ۹: مدل شبکه عصبی مورد استفاده برای طبقه بندی تورم های قطعات کامپوزیتی

همانطور که در جدول ۲ می بینید لایه آخر مدل ما از Softmax استفاده شده است که بتوان احتمال وجود هر لایه را به صورت درصد بیان می کند که مجموع درصد ۱۳ حالت طبقه بندی شده ۱۰۰ درصد می باشد.

دقت خروجی این روش به 90.1 درصد می رسد که دقت بسیار خوبی می باشد ولی با این عیب که تنها محل تورم پیدا می شود و ما هیچ اطلاعی از اندازه تورم نخواهیم داشت. ولی مدل شبکه بسیار سبکی دارد با پارامتر های قابل آموزش کمی همراه است و این نشان دهنده نیاز نداشتن به GPU بسیار قوی است و همینطور حجم محاسباتی بسیار کمی دارد به دلیل اینکه حداکثر تعداد فیلتر هایی که در اختیار داریم ۵۱۲ می باشد که باید کانو ۳ در ۳ روی آن اعمال بشود که با تجربه می توان گفت Memory GPU کمتر از ۳ گیگ یا حتی ۲ گیگ لازم است.

Layer Name	Layer Description
Input	256×256 2-channel FFT Spectrum Image
Convolution 1, Pooling 1	Convolution Filter 3×3, strides 1, Number of filter = 16, ReLU, Batch normalization, Max Pooling Filter 2×2, strides 2
Convolution 2, Pooling 2	Convolution Filter 3×3, strides 1, Number of filter = 32, ReLU, Batch normalization, Max Pooling Filter 2×2, strides 2
Convolution 3, Pooling 3	Convolution Filter 3×3, strides 1, Number of filter = 64, ReLU, Batch normalization, Max Pooling Filter 2×2, strides 2
Convolution 4, Pooling 4	Convolution Filter 3×3, strides 1, Number of filter = 128, ReLU, Batch normalization, Max Pooling Filter 2×2, strides 2
Convolution 5, Pooling 5	Convolution Filter 3×3, strides 1, Number of filter = 256, ReLU, Batch normalization, Max Pooling Filter 2×2, strides 2
Convolution 6, Pooling 6	Convolution Filter 3×3, strides 1, Number of filter = 512, ReLU, Batch normalization, Max Pooling Filter 2×2, strides 2
Fully Connected	Input = 8,192, Output = 2,024, ReLU
Softmax	Input = 2,024, Output = 13

جدول ۲: مدل شبکه عصبی با توضیحات کامل از هر لایه آن



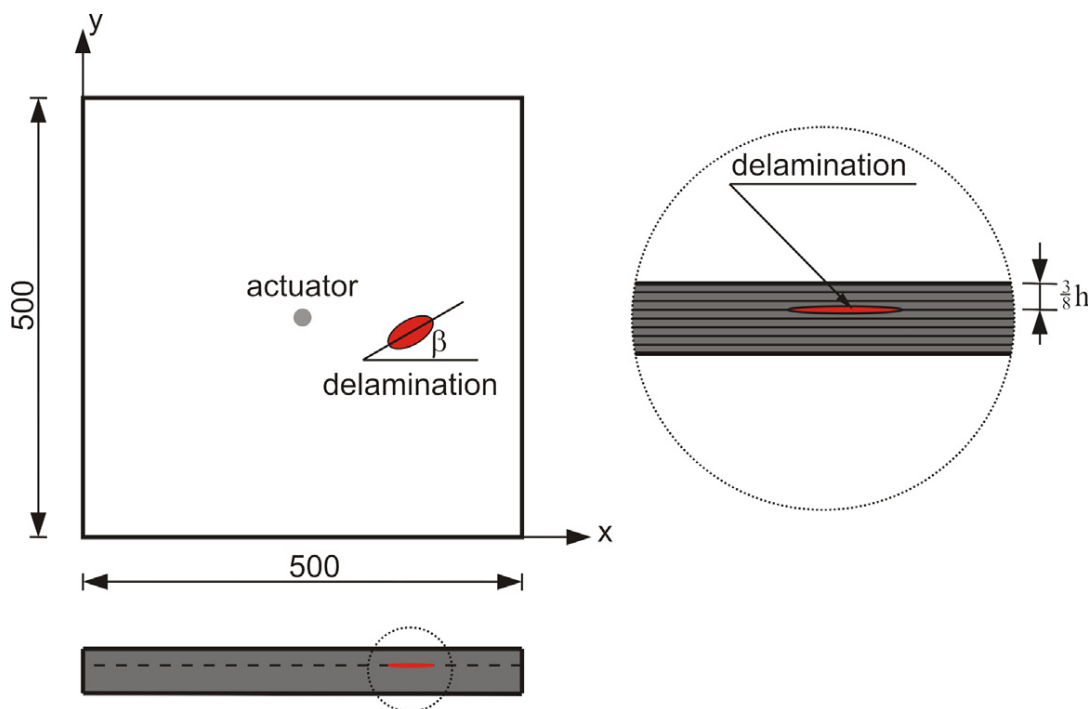
شکل ۱۰: نمونه ای از تولید دیتا های گرفته شده که بصورت فریم های زمانی از اندازه فرکانسی کل صفحه بالا و پایین قطعه مد نظرمون می باشد.

ii. Full wavefield processing by using FCN ...

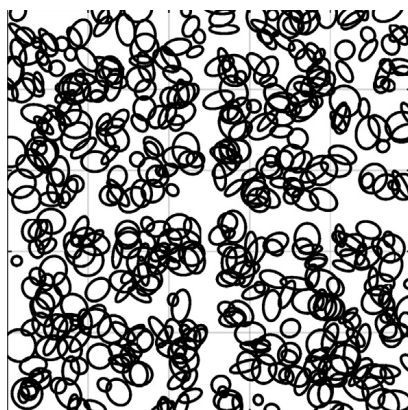
روش full wavefield به صورت موازی طیف رنگی المان های ارتعاش داده شده صفحه کامپوزیت ها^{۱۵} مورد بررسی قرار میدهد. طبق مطالب گفته شده از تولید طیف رنگی به کمک الگوریتم STFT می توان اینگونه این مطلب رو توجیح نمود که به صورت بی نهایت سنسور پیزوالکتریک را روی صفحه قطعه کامپوزیتیمون داریم که با دریافت داده های فرکانسی در طول زمان مشخص می توان فریم هایی به تعداد پله های زمانی انتخاب شده تولید کرد که فرکانس کل قطعه مد نظر با طیف رنگی که بیانگر اندازه آن هاست مشخص می شود. از الان می توان دلیل برتری این روش را نسبت به روش گذشته بیان نمود چرا که داده های ورودی شبکه ما به صورت ۳ بعدی می باشد. می توان تور نمود که گرفتن فریم های گفته شده به صورت موازی از کل صفحه قطعه کامپوزیتی چقد عملیات سنگینی باشد بعلاوه اینکه در مقاله گفته شده به تعداد ۴۷۵ نمونه تورم از قطعات مختلف تولید کرده و برای هرکدام از این ۴۷۵ نمونه به تعداد ۳ فریم زمانی به صورت موازی فرکانس طبیعی کل قطعه را دریافت کرده که طبق گفته خود مقاله اینکار با کارت گرافیک Tesla K20X GPU انجام شده و به مدت ۳ ماه دیتا برداری انجام شده است چرا که این عملیات باید برای هر رندم ارتعاشات هارمونیک مختلف به مدت ۱۰۰۰ بار انجام شود.

^{۱۵} روش انجام اینکار طبق مقاله به صورت زیر انجام پذیر می باشد:

It resembles a full wavefield measurements acquired on a surface of the plate by the scanning laser Doppler vibrometer.



شکل ۱۱: نمونه ای از تورم وارد شده از ۴۷۵ نوع تورم گفته شده می باشد که عمق و ابعاد تورم در شکل مورد نمایش داده شده است.



شکل ۱۲: شکل کلی از ۴۷۵ نمونه تورم که در یک قطعه نمایش داده شده است.

می توانید نمونه ای از قطعه دیتای ورودی را در شکل ۱۱ ببینید و همینطور کل نمونه هایی از قطعات دیتای ورودی را در شکل ۱۲ مشاهده فرمایید. توجه داشته باشید که هر نمونه قطعه باید برای ۱۰۰۰ نمونه ارتعاشات هارمونیک دیتا برداری شود که در این صورت دیتای بسیار سنگینی را در اختیار ما می گذارد.

۱.ii. پیش پردازش

در این مقاله علاوه بر پیش پردازش های گفته شده برای تولید فریم های طیف های رنگی قطعه، روشی برای کاهش کانال های دیتای ورودی به شبکه عصبی را ارائه نموده است چرا که با هر چه تعداد کانال های ورودی ما بیشتر باشد حجم محاسباتی ما هم بیشتر شده و باعث محدودیت



شکل ۱۳: نمونه ای از خروجی فریم های صفحه بالا و پایینی قطعه کامپوزیتی.

هایی در مموری کارت گرافیکی ما ایجاد می کند. این روش را می توان با تابع RMS پیاده نمود که به صورت زیر تعریف می شود:

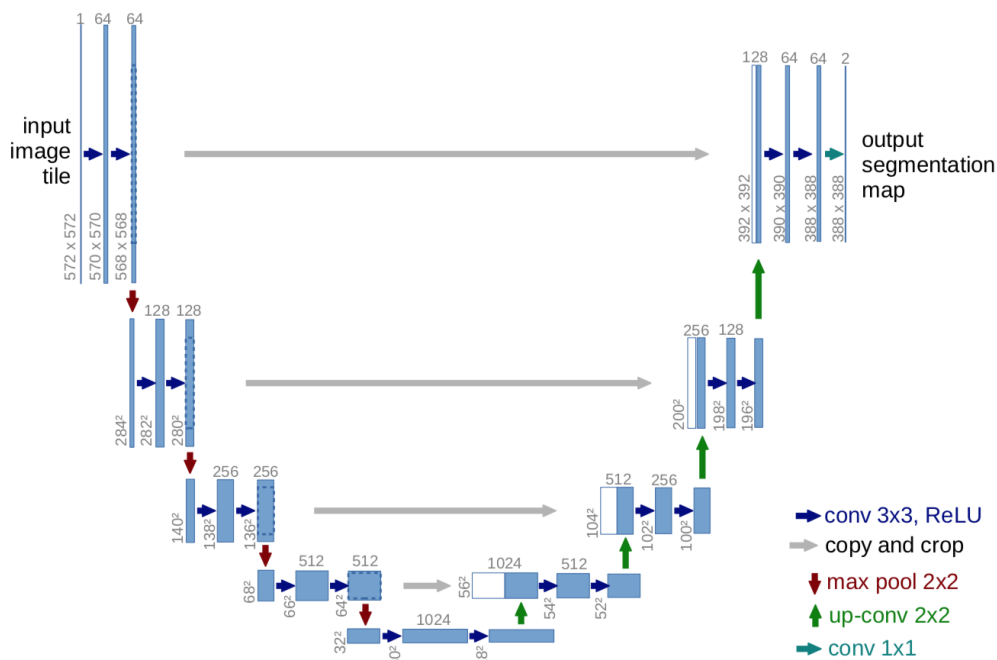
$$\hat{s}(x, y) = \sqrt{\frac{1}{N} \sum_{k=1}^N s(x, y, t_k)^2} \quad (4)$$

که s اندازه فرکانس المان x, y در زمان t_k می باشد که با مجذور فریم های پله زمانی در هر قطعه تبدیل به \hat{s} هایی می شود که تنها تابعی از x, y می باشد.

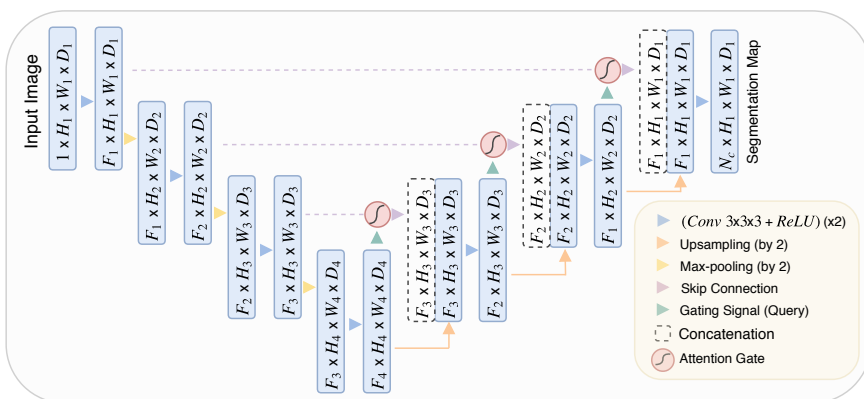
۲.ii. پیدا کردن محل و اندازه تورم با روش یادگیری عمیق

پس دیتای ورودی ما دو عکس ۵۱۲ در ۵۱۲ (عرض و طول قطعه را به ۵۱۲ قسمت تقسیم کردیم) که شامل صفحه بالا و پایین قطعه کامپوزیت می باشد در نظر میگیریم و با کمک مدل های شبکه عصبی تقسیم بندی^{۱۶} که شامل Unet می باشد اندازه و محل تورم را پیدا کنیم. مدل پایه اینکار را می توانید در شکل ۱۴ ببینید ولی این می توان این مدل را بهینه و قوی تر نمود که ابتدا به آن Attention اضافه نمود که در شکل ۱۵ معماری آن قابل مشاهده می باشد. و علاوه بر این می توان با اضافه کردن بلوک های Residual و Recurrent بجای کانو های ۳ در ۳ پشت سر هم، مدل شبکه را قوی تر نمود (شکل ۱۶ معماری مدل نهایی تقسیم بندی به کمک Unet می باشد). شکل گراف مدل بسیار سنگین می باشد برای همین قابلیت گذاشتن آن در این فایل پی دی اف وجود ندارد و می توانید مدل را در همین فایل (figs/Final Model with more details.png) یا در لینک [Final Model with more details](#) دریافت نمایید.

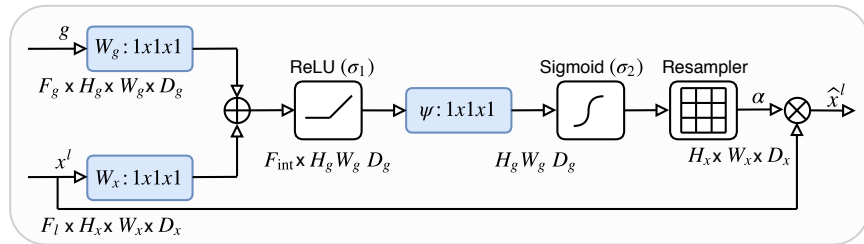
^{۱۶} Segmentation



شکل ۱۴: معماری مدل پایه تقسیم بندی U-net

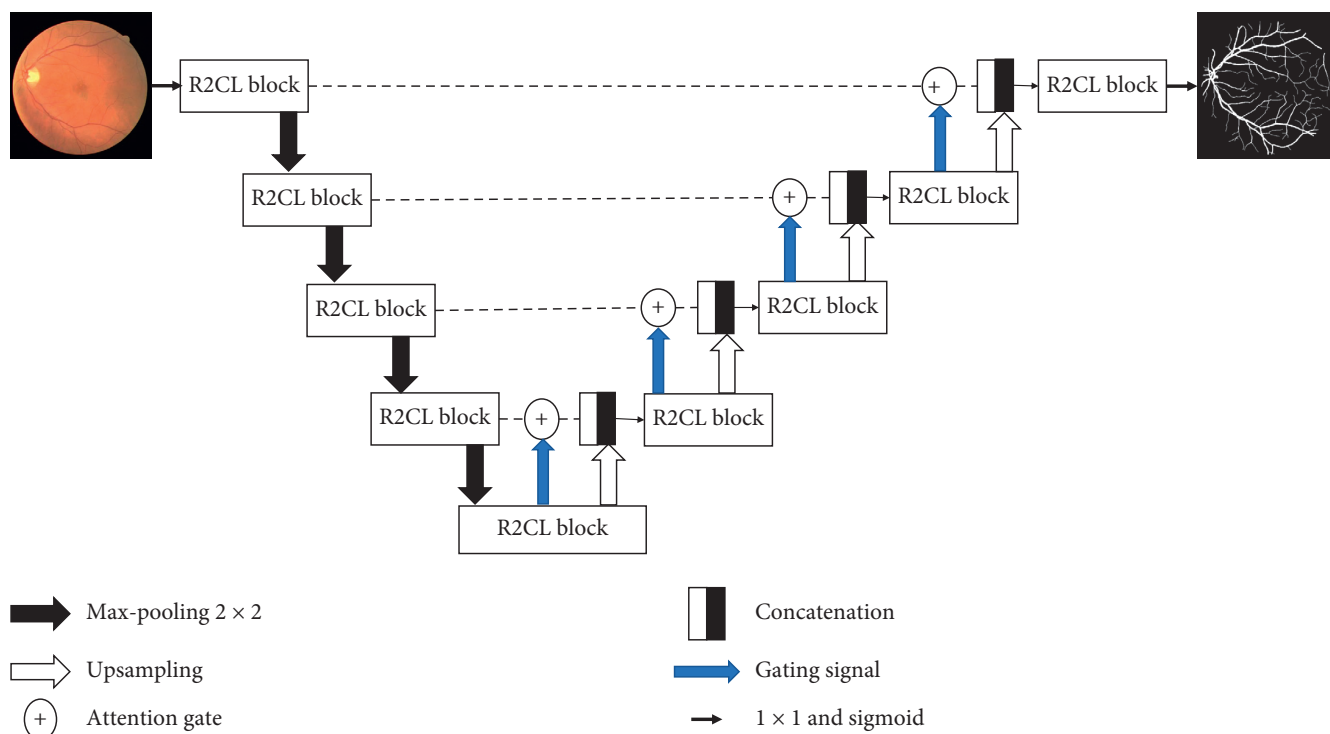


(ا) نمایش U-net با Attention هایی برای قوی تر نمودن توجه مدل به محل تورم

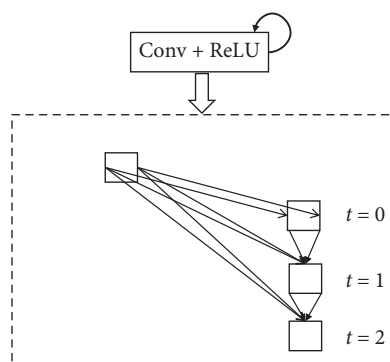


(ب) جزئیات بلوک های اضافه شده به مدل شبکه یونت

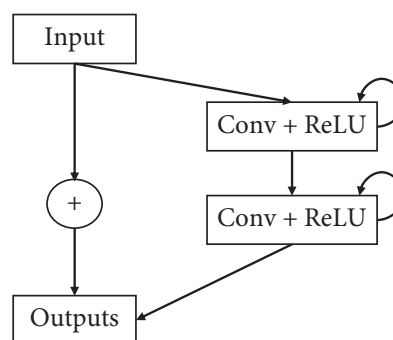
شکل ۱۵: مدل شبکه معماری U-net با Attention که شامل بلوک هایی است که در شکل آ می توانید محل این بلوک ها را مشاهده کنید و جزئیات این بلوک در شکل ب مورد نمایش گذاشته شده است.



(ا) شکل کلی از مدل نهایی R2AU-Net



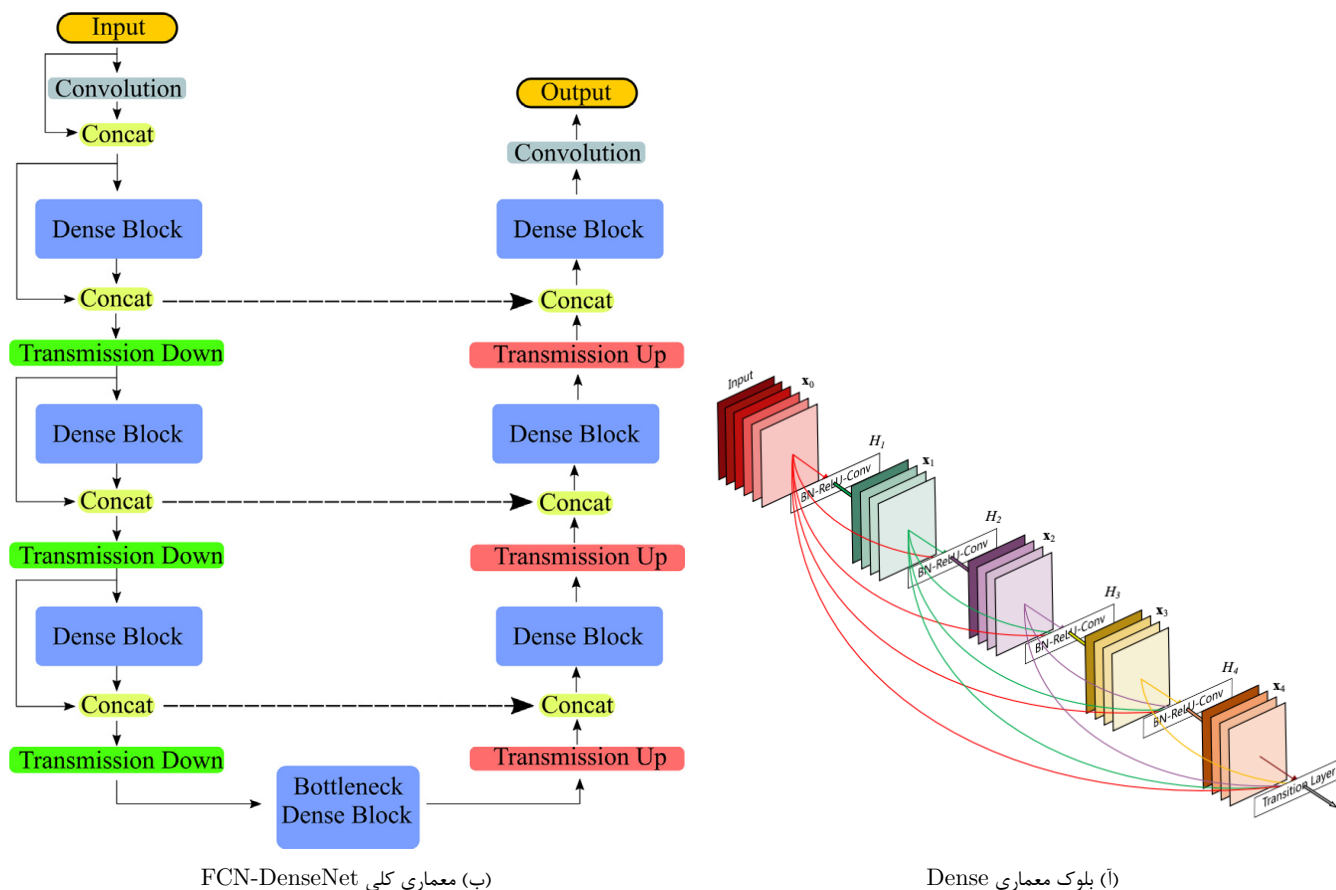
(ج) نمایش بلوک های Recurrent



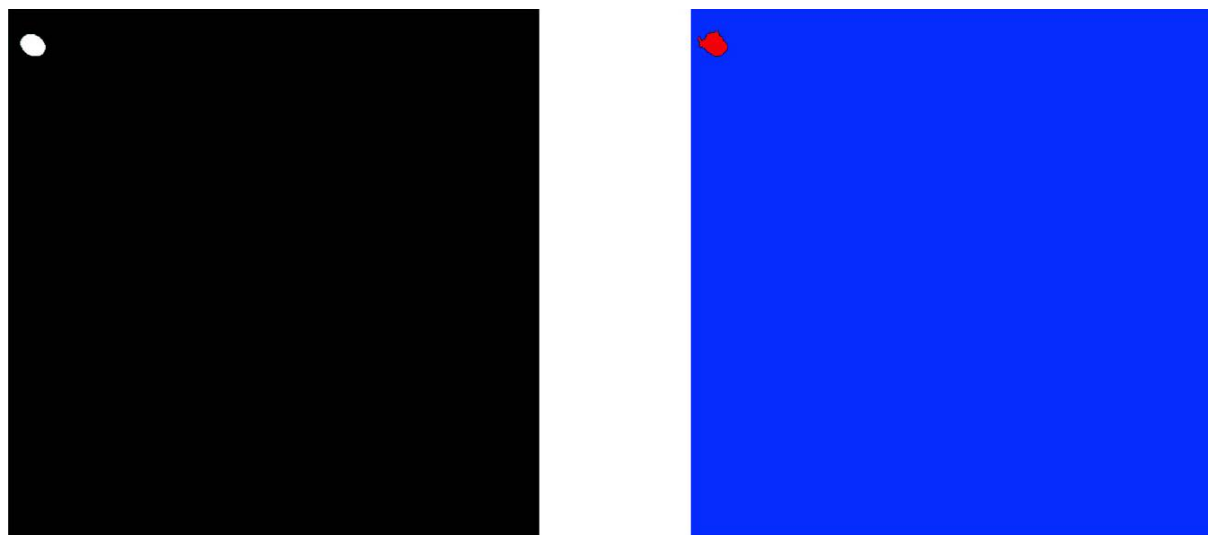
(ب) جزئیاتی از بلوک های R2CL

شکل ۱۶: Attention Recurrent Residual Convolutional Neural Network

ولی در این مقاله از معماری FCN-DenseNet استفاده کرده است که شبیه به همان مدل U-net می باشد ولی با این تفاوت که دارای بلوک های Dense می باشد. که بلوک های DenseNet را می توانید در شکل ۱۷ ببینید و همینطور کل معماری در شکل ۱۷ ب گذاشته شده است. نمونه ای از خروجی این مدل را در شکل ۱۸ می توانید مشاهده کنید. این مدل پیاده سازی شده و می توانید گراف آن را در لینک [FCN-DenseNet architecture](#) ببینید



شکل ۱۷: معماری شبکه گفته شده در این مقاله با هدف تقسیم بندی عکس خروجی برای رسیدن به اندازه و محل تورم.



شکل ۱۸: مقایسه خروجی مدل و واقعی شکل و محل تورم. عکس سمت راست خروجی مدل می باشد و عکس سمت چپ تورماعمال شده واقعی.

بیشترین دقت این مدل تا به مقدار اشتراک به اجتماع ۹۳ درصد رسیده است که می توان دقت قابل قبولی دانست ولی با توجه به مشکل بودن جمع آوری دیتای مد نظر محدودیت هایی را ایجاد می کند.