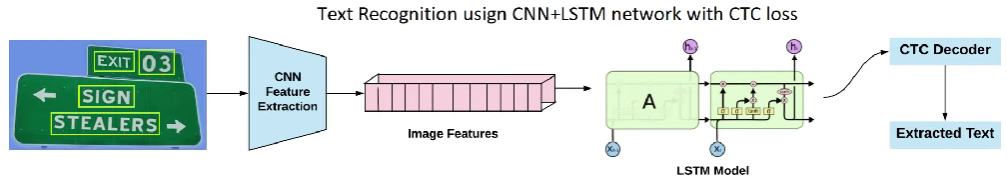
علاوه بر چالش هایی که خود ocr داره زبان فارسی و عربی هم چالش های بیشتری داره از جمله حروف به هم چسبیده و کمبود دیتاست. حالا در مرحله اول میایم و درباره روش های مختلف صحبت می کنیم:

یکی از ساختارهایی که برای text recognization استفاده می کنند:ببین اون LSTM پس از CNN باعث میشه که علاوه بر اینکه به صورت منفرد بتونه هر کاراکتر تشخیص بده باعث بشه که اون معانی بین کلمات رو هم در نظر بگیره. و برای loss نیز از CTC استفاده می کنه.



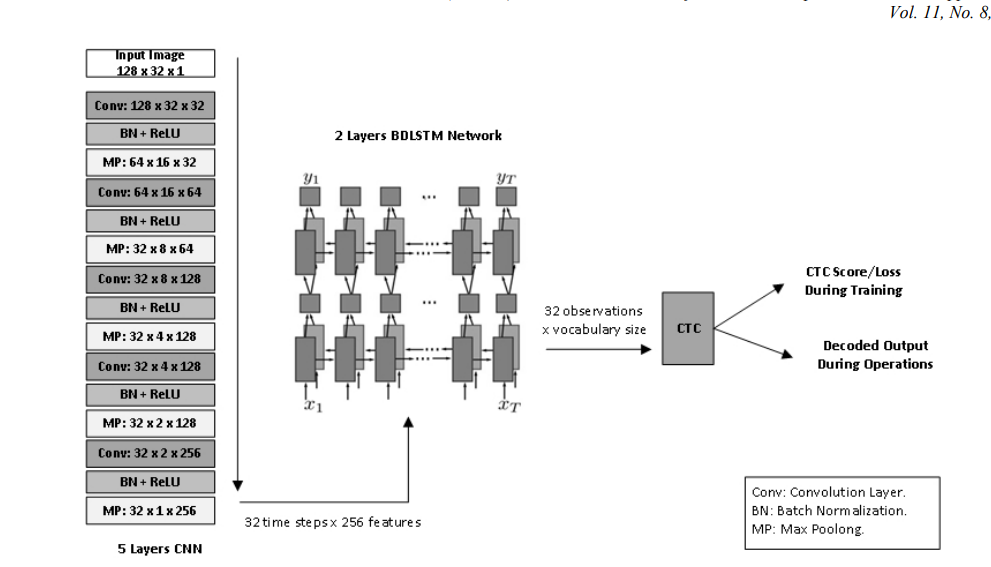
**CTC loss چی هست؟**

ما در ابتدا یک لیستی از کاراکترهایی که داریم رو به مدل می دهیم و بعد یک ماتریسی تشکیل میشه چقدر احتمال داره فلان کاراکتر ، کاراکتر اول باشه دوم باشه و .... دیگه وارد مراحل backward و اپدیت کردنش نمیشم ولی چیز کلیش همینه.

کلا برای ocr مراحل زیر را می توانیم طی کنیم:

1. افزایش کیفیت اسناد (میشه نویز کاهش داد ابزاری تو opencv برای این کار هستند)
2. تشخیص مناطقی که درون آن ها متن هست.
3. استخراج ویژگی ها از اون مناطق feature extraction
4. اجرای فرایند ششناسایی کاراکترها

مقاله ای هم که مطالعه کردم روشی مشابه موارد بالا پیشنهاد کرده است:



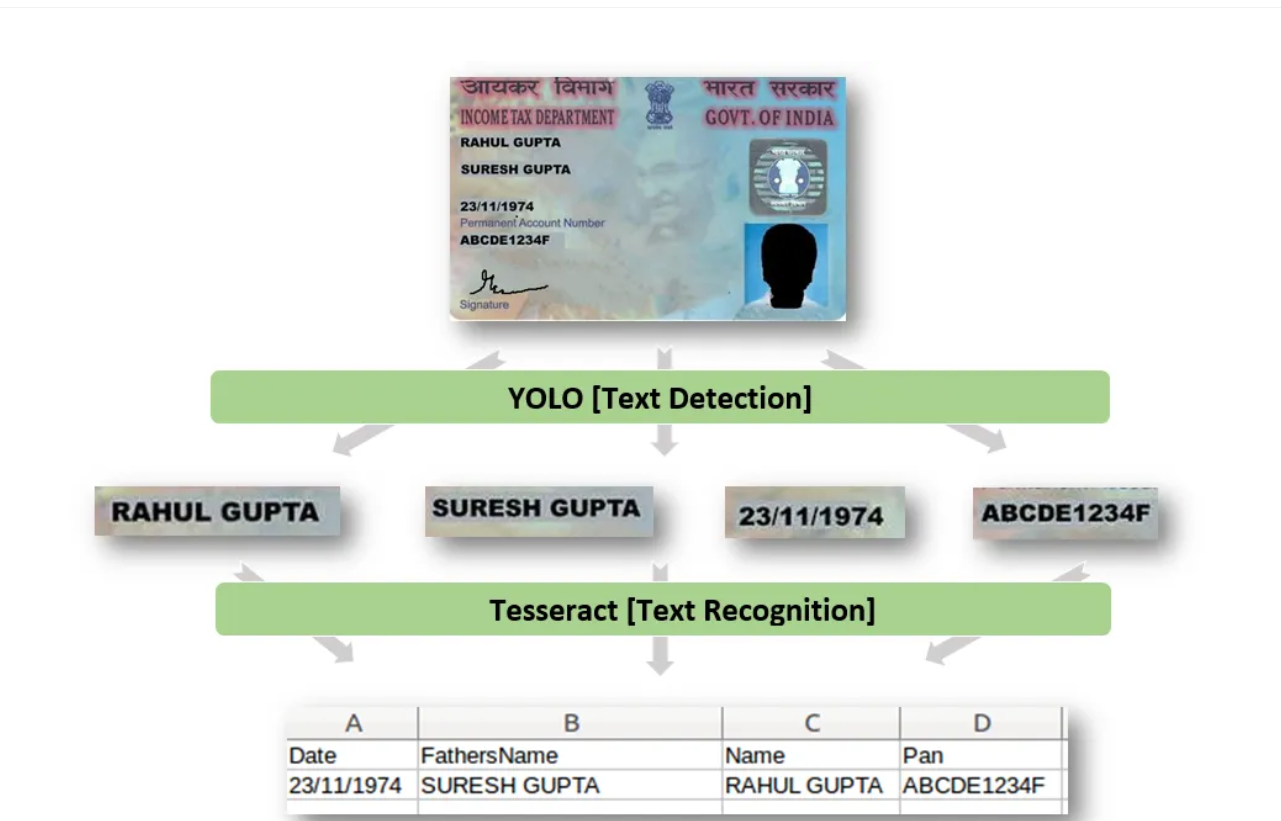
یه چالشی که برای تصویر که از نوشته داریم نیاز باشه برش بزنیم مثل یک scanerr میتونیم اینجا از یکسری از لایبرری های opencv استفاده بکنیم می توانیم از findcountour استفاده بکنیم و بعد با cv2.approxPolyDP میتونیم گوشه های تصویر پیدا کنیم یعنی همون corner ها و بعد با یک cv2.getPerspectiveTransform ماتریس transform پیدا می کنیم و در نهایت با wrapPerspective میایم و تصویر چرخش میدیم:







اگه مثلا از این مدل های ocr می خوایم در خوندن حروف کارت ملی و بانکی استفاده بکنیم استفاده از YOLO که یکی از الگوریتم های object detection هست میتونه خیلی خوب باشه



در یکی از ریپو ها که برای ocr فارسی داشتم نگاه میکردم خودش یک دیتاست فارسی ساخته بود از کلمات وبعد میومد اون تصویر یک threshold میزد که فاصله رنگی میان پس زمینه و نوشته زیاد بشه و بعد یک dilate میزد که کلمات به هم بچسبند که بتونه همون cv2.countour بزنه و بعد که دور هر کلمه میومد و اون rectangle می کشید.

