The need for handwritten recognition

As we all know that when children start to write, their formation of characters are not as accurate, and thus it would be interesting to see if a model can be built to approximate what character the child is trying to write.

According to the paper Rabko et al. handwriting recognition can be used on a touch screen device to allow children to write answers to math questions presented to them. This will encourage the children to do math exercises, moreover it will indicate to the user the accuracy of the character being drawn and the user can determine whether the character predicted by the model is the same character the child is trying to input into the device. Additionally, the model will force the child to write accurately as possible to obtain the correct character the user wishes to input.

Norman’s Model of Interaction

Before norman’s model is actually executed, the application on the touch screen device has to display the question on the screen to the user. For instance, “”.

1. Establishing the goal.

* At this stage, the child (user) should have calculated the answer, and thus is stored in his/her memory temporarily.
* The goal of the user should be to transfer the answers’ mental representation to the device.

1. Forming the intention.

* The user should have an understanding on how to successfully achieve the goal.
* A more specific intention should be formulated, which is to write the answer to the question on the screen. This can be done by a stylus or the user’s finger.

1. Specifying the action sequence.

* At this stage, the user should have formulated a sequence of actions to accomplish the task.
* Such a sequence could be: determine the answer 🡪 have a mental representation of the answer 🡪 form the intention of writing the answer on the screen.

1. Executing the action.

* The formulated intention is turned into a practical action, and now the goal should be complete.

At this stage, the system performs its task which is to convert the handwritten answer to text by approximating what the written character could be. This is done by paying attention to the strokes, patterns, etc. and classifies the character based on what the model has been trained on.

1. Perceiving the system state.

* The user waits for the system to complete the task.
* Once the task is completed, the user perceives the action, that is the user witnesses the result of the output.

1. Interpreting the system state

* At this stage, the system should have completed its processing and is ready to communicating the computation result to the user.
* The system provides the output to the user, and the user views the output and interprets the output by comparing it to the desired goal of the user.

1. Evaluating the system state with respect to the goals and intentions.

* Firstly, based on the user’s intention the user evaluates whether the intention was correct. Based on whether the intention was correct or not, but the output is not as expected, then this could imply two things:

1. The user does not know how to use the system in the required manner to obtain the desired result. Basically, the user is performing a slip in which he/she is forming the intention correctly, however the action is not performed correctly.   
   So in this case, the child wrote a digit but the system classified the written digit incorrectly, hence the child should try to write the digit again as accurately as possible.
2. The user is forming the incorrect intention; hence the incorrect output is received.  
   In this case, the child could be either scribbling or drawing something. Basically, the child is not concentrating or does not know how to write the character as accurately as possible, maybe due to the fact that he/she does not know how to form the character, the child may be inexperienced, the child is too young, etc.

* Over here, if the desired output is not being received the user can decide one of two actions; the user can try to run the process all over again or the user can refine the goal with a new intention and try to execute the new intention by starting the entire process again.
* If the output matches the result desired by the user, the application