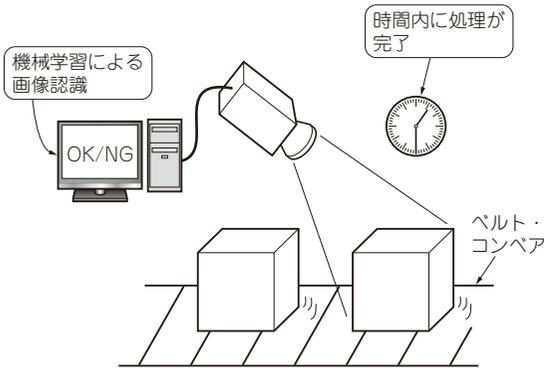
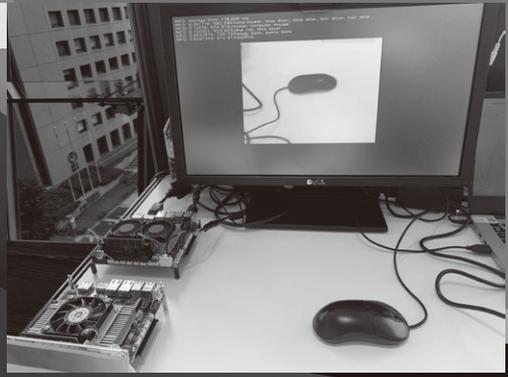
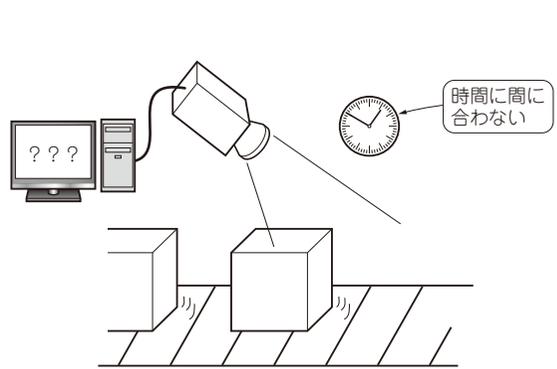


# TensorFlow Lite × RTOSで 機械学習の リアルタイム 処理実験

大貫 良一



(a) 荷物が流れる前に画像認識処理を終えることができた



(b) 規定時間以内に画像認識処理が終わらなかったため荷物が流れていってしまった

図1 リアルタイム処理が必要な場合…工場のラインでの機械学習による画像認識

## 機械学習がx86のリアルタイム環境で動くようになった

機械学習のフレームワークとして有名なTensorFlowおよびTensorFlow Lite (いずれもGoogle)は、ほとんどの場合LinuxやAndroid, iOS, ベアメタルのマイコンなど、非リアルタイム環境で使われています。そのため、機械学習を使った推論エンジンは、実行時間が保証されていない非リアルタイム環境での使用にとどまり、リアルタイム制御が必要な処理には使用できませんでした(図1)。実際の現場では、リアルタイムOSを使いつつ、機械学習、特に物体、人物、不良品、問題動作の判定精度を向上させ、製品性能や不良率を改善したいという需要が数多くあります。

このような用途では、グラフィックやネットワーク・スタック、スクリプト・エンジンなどの様々なアプリケーションが1台のボードで動く必要があります。そのためには高速なCPUが必要になり、x86 CPUが最適になります。産業用PCを使った産業用表示機や制御装置などが1つの例になります。

そんなとき、x86の産業用PCでも動作するリアルタイムOSであるVxWorks(ウインドリバー)が、2021年7月リリースのVxWorks 7 21.07よりTensorFlow Liteをサポートしました。これによりx86のリアルタ

イム環境で機械学習の結果を使うことが可能になりました。TensorFlow Liteは機械学習の推論/判定のみを行い、機械学習の訓練はPC上でTensorFlowを使うことで軽量化したソフトウェアです(図2)。

ここでは、TensorFlow Liteを使った実験を行い、高負荷時での推論エンジンの処理時間を計測し、リアルタイム環境であるVxWorksと非リアルタイム環境であるLinuxでの機械学習の動作の違いを比較します。

## リアルタイムOSと非リアルタイムOSを比較する

### ● 実験の方針

VxWorksとLinuxで同じアプリケーションを動作

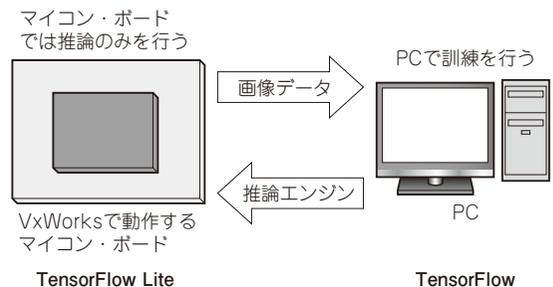


図2 TensorFlow LiteとTensorFlowの関係